Academic Regulations
Programme Structure
and
Detailed Syllabus

Bachelor of Technology (B.Tech)
in
Computer Science and Business System

(Four Year Regular Programme)

(Applicable for Batches admitted from 2025-26)

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous)

Bachupally, Kukatpally, Hyderabad- 500 090

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

HYDERABAD

Academic Regulations for B.Tech (Regular) under GR25

(Applicable for Batches Admitted from 2025-26)

Under Graduate Degree Programme in Engineering and Technology (UG)

Gokaraju Rangaraju Institute of Engineering and Technology (GRIET) offers a 4-year (8 Semesters) Bachelor of Technology (B.Tech) degree programme. The following programmes are offered in GRIET.

S.No	Department	Programme Code	Programme
1	Civil Engineering	01	B.Tech Civil Engineering
2	Electrical and Electronics Engineering	02	B.Tech Electrical and Electronics Engineering
3	Mechanical Engineering	03	B.Tech Mechanical Engineering
4	Electronics and Communication Engineering	04	B.Tech Electronics and Communication Engineering
5	Computer Science and Engineering	05	B.Tech Computer Science and Engineering
6	Computer Science and Business System	32	B.Tech Computer Science & Business System
7	Computer Science and Engineering (AIML)	66	B.Tech Computer Science and Engineering (Artificial Intelligence & Machine Learning)
8	Computer Science and Engineering (Data Science)	67	B.Tech Computer Science and Engineering (Data Science)

GR25 Regulations shall govern the above programmes offered by the Departments with effect from the students admitted to the programmes in 2025-26 academic year is given below.

1. **Medium of Instruction:** The medium of instruction (including examinations and reports) is English.

2. **Admissions:** Admission to the undergraduate (UG) Programme shall be made subject to the eligibility, qualifications and specialization prescribed by the Telangana State Government/JNTUH University from time to time. Admissions shall be made either on the basis of the merit rank obtained by the student in the common entrance examination conducted by the Government/University or on the basis of any other order of merit approved by the Government/University, subject to reservations as prescribed by the Government/University from time to time.

3. **B.Tech Programme Structure**

3.1 A student after securing admission shall complete the B.Tech programme in a minimum period of four academic years and a maximum period of eight academic years starting from the date of commencement of first year first semester, failing which student shall forfeit seat in B.Tech course. Each student has to secure a minimum of 160 credits out of 164 credits for successful completion of the undergraduate programme and award of the B.Tech degree.

3.2 **UGC/ AICTE** specified definitions/ descriptions are adopted appropriately for various terms and abbreviations used in these academic regulations/ norms.

3.2.1 Semester Scheme

The undergraduate programme is of four academic years and there shall be two semesters in each academic year. There shall be a minimum of 15 weeks of instruction, excluding the mid-term and semester-end exams. Around 15 instruction hours, 30 instruction hours and 45 hours of learning need to be followed per one credit of theory course, practical course and project/field-based learning respectively. In each semester, there shall be ‘Continuous Internal Evaluation (CIE)’ and ‘Semester End Examination (SEE)’ under Choice Based Credit System (CBCS).

3.2.2 Credit Courses

All courses offered in each semester are to be registered by the student. Against each course in the course structure, the L: T: P: C (lecture periods: tutorial periods: practical periods: credits) pattern has been defined.

- One credit is allocated for one hour per week in a semester for lecture (L) or Tutorial (T) session.
- One credit is allocated for two hours per week in a semester for Laboratory/ Practical (P) session.
- One credit is allocated for three hours per week in a semester for Project/Mini-Project session.

For example, a theory course with three credit weightage requires three hours of classroom instruction per week, totaling approximately 45 hours of instruction over the entire semester.

3.2.3 Subject Course Classification

All subjects/courses offered for the undergraduate programme in E&T (B.Tech degree programmes) are broadly classified as follows.

S. No.	Broad Course Classification	Course Group/ Category	Course Description
1	BS	Basic Sciences	Includes Mathematics, Physics and Chemistry courses
2	ES	Engineering Sciences	Includes Fundamental Engineering Courses
3	HS	Humanities and Social Sciences	Includes courses related to Humanities, Social Sciences and Management
4	PC	Professional Core	Includes core courses related to the parent branch of Engineering
5	PE	Professional Electives	Includes elective courses related to the parent branch of Engineering
6	OE	Open Electives	Elective courses which include inter-disciplinary courses or courses in an area outside the parent branch of Engineering
7	PC	Project Work	B.Tech Project Work
8	PC	Industry Training/ Internship/ Industry Oriented Mini- project	Industry Training/ Internship/ Industry Oriented Mini-Project
9	PC	Seminar	Seminar based on core contents related to parent branch of Engineering
10	SD	Skill Development Courses	Courses designed to help individuals gain, improve, or refine specific skills
11	VAC	Value Added Courses	Courses to build professional values, traditional knowledge and sensitization of societal issues

4. Mandatory Induction Programme

An induction programme of one week duration for the UG students entering the institution, right at the start shall be implemented. Normal classes commence only after the induction programme is conducted. Following activities could be part of the induction programme: i) Physical Activity ii) Creative Arts iii) Imparting Universal Human Values iv) Literary Activities v) Lectures by Eminent People vi) Visits to Local Areas vii) Familiarization to department as well as entire institute and viii) Making students understand Innovative practices at the college premises etc.

5. Course Registration

5.1 A faculty advisor / mentor shall be assigned to a group of around 20 students, who will advise the students about the undergraduate programme, its course structure and curriculum, choices/options of the courses, based on their competence, progress, pre-requisites and interest.

5.2 A student shall register for all the courses offered in a semester as specified in the course structure. Course registrations are exercised through F-235 form.

5.3 **Professional Electives:** The students have to choose six Professional Electives (PE-I to PE- VI) from the specified list.

Students have the flexibility to choose from the list of professional electives offered by the Institute or opt to register for the equivalent Massive Open Online Courses (MOOCs).

5.4 **Open Electives:** Students have to choose three Open Electives (OE-I, II & III) from the two threads of Open Electives given by other than the parent department. However, the student can opt for an Open Elective course offered by his parent department, if the student has not studied that course so far. Similarly, Open Elective courses being studied should not match with any courses of the forthcoming semesters.

Students have the flexibility to choose from the list of open electives offered by the Institute or opt to register for the equivalent Massive Open Online Courses (MOOCs).

5.5 **Provision for Early Registration of MOOCs:**

For a professional elective/ open elective in a semester, students are allowed to register for an equivalent MOOCs course listed from time to time by the University one semester in advance. For example, a Professional Elective of III Year II Sem shall be allowed to register under MOOCs platform in III year I Sem.

The credits earned in one semester in advance can be submitted in the subsequent semester for the assessment.

The students who have registered in advance in an equivalent MOOCs course and fail to secure any pass grade in the MOOCs course, can register for the regular course offered in the following semester of their course structure.

5.6 **Conversion of Marks Secured in MOOCs into Grades:** Marks secured in the internal and external evaluations of a MOOCs course shall be scaled to 40 and 60 marks respectively. The sum of these two components shall be considered as the total marks out of 100. The corresponding grade shall then be determined as per the marks-to-grades conversion rules specified in Clause 10.3.

5.7 MOOCs are allowed only for either PE-I, PE-II or OE-I, OE-II courses and for few Minors & Honors courses

5.8 **Additional learning resources:**

Students are encouraged to acquire additional course-related knowledge by auditing learning resources from MOOCs platforms for each course offered in their course structure. These additional courses are not meant for earning credits but are intended to enhance knowledge.

6. Rules to offer Elective courses

6.1 An elective course may be offered to the students, only if a minimum of 25% of class strength opts for it.

6.2 Same elective course for different sections may be offered by different faculty members. The selection of elective course by students will be based on first come first serve and / or CGPA criterion.

6.3 If the number of students registrations are more than the strength of one section, then it is choice of the concerned Department to offer the same course for more than one section based on the resources available in the department.

7. Attendance requirements:

7.1 A student shall be eligible to appear for the semester-end examinations, if the student acquires a minimum of 75% of aggregate attendance of all the courses for that semester.

7.2 Shortage of attendance in aggregate upto 10% (securing 65% and above but below 75%) in each semester may be condoned by the college academic committee on genuine and valid grounds, based on the student's representation with supporting evidence.

7.3 A stipulated fee shall be payable for condoning of shortage of attendance as notified in the respective college websites.

7.4 Two hours of attendance for each theory course shall be considered, if the student appears for the mid-term examination of that course.

7.5 Shortage of attendance below 65% in aggregate shall in no case be condoned.

7.6 Students whose shortage of attendance is not condoned in any semester, are not eligible to take their semester-end examinations of that semester. They get detained and will not be promoted to the next semester. Their registration for that semester shall stand cancelled, including internal marks. They may seek re-registration for that semester in the next academic year.

7.7 A student fulfilling the attendance requirement in the present semester shall not be eligible for readmission into the same semester

8. Criteria for Earning of Credits in a Course

8.1 A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each course, if the student secures not less than 35% (21 marks out of 60 marks) in the semester end examinations (SEE), and a minimum of 40% (40 marks out of 100 marks) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together; in terms of letter grades, this implies securing 'C' grade or above in that course.

8.2 A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to Field Based Research Project / Industry Oriented Mini Project / Internship, if the student secures not less than 40% marks (i.e. 40 out of 100 allotted marks) in each of them. The student is deemed to have failed, if he/she (i) does not submit a report on Field-Based Research Project/Industry Oriented Mini Project/ Internship, or (ii) not make a presentation of the same before the evaluation committee as per schedule, or (iii) secures less than 40% marks in Field- Based Research Project / Industry Oriented Mini Project / Internship evaluations.

8.3 A student eligible to appear in the semester-end examination for any course, is absent from it or failed (thereby failing to secure 'C' grade or above) may re-appear for that course in the supplementary examination as and when it is conducted. In such cases, internal marks assessed in continuous internal evaluation (CIE) earlier for that course will be carried over, and added to the marks obtained in the SEE supplementary/make-up examination. If the student secures sufficient marks for passing, 'C' grade or above shall be awarded as specified in clause 10.3.

9. Distribution of Marks and Evaluation

9.1 The performance of a student in every course (including Value Added Courses and Skill Development Courses, Laboratory/Practical and Project Work) will be evaluated for 100 marks each, with 40 marks allotted for CIE (Continuous Internal Evaluation) and 60 marks for SEE (Semester End-Examination), irrespective of the credits allocated.

9.2 Continuous Internal Evaluation (CIE)

9.2.1 Theory Courses:

For theory courses, during a semester, there shall be two mid-term examinations. Each Mid-Term examination consists of two parts i) Part – A for 10 marks, ii) Part – B for 20 marks, totaling to 30 marks. Total duration of mid-term examination is two hours.

1. Mid Term Examination for 30 marks:
   a. Part - A : Objective/quiz paper for 10 marks.
   b. Part - B : Descriptive paper for 20 marks.

The objective/quiz paper is set with multiple choice, fill-in the blanks and match the following type of questions for a total of 10 marks.

The descriptive paper shall contain 6 questions out of which, the student has to answer 4 questions, each carrying 5 marks. The average of the two Mid Term Examinations shall be taken as the final marks for Mid Term Examination (for 30 marks).

While the first mid-term examination shall be conducted on 50% of the syllabus, the second mid-term examination shall be conducted on the remaining 50% of the syllabus. Questions will be drawn from the mid-term exam syllabus, ensuring uniform coverage of all topics.

The remaining 10 marks of Continuous Internal Evaluation are distributed as follows:

2. Five marks for the assignment for 5 marks. Student shall submit two assignments and the average of 2 Assignments each for 5 marks shall be taken. The first assignment should be submitted before the conduct of the first mid-term examination, and the second assignment should be submitted before the conduct of the second mid-term examination.

3. Five marks for the Quiz/Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the concerned subject. This assessment shall be completed before II Mid-Term Examination.

9.2.2 Graphics for Engineers Course:

For this course, 20 marks will be allocated for day-to-day assessments conducted during drawing practice sessions, and another 20 marks will be allocated for the mid-term examination. In the mid-term examination, students shall attempt any four out of six given questions. Each mid examination is conducted for 90 minutes. Average of the two mid exams shall be considered.

9.3 Computer-Based Test (CBT) in each course is available for students who either:

1. missed one of the two mid-term examinations due to unavoidable circumstances, or
2. attended both mid-term examinations but wish to improve their internal marks.

The CBT will be conducted at the end of the semester and will carry a total of 30 marks. The marks obtained in the CBT will be considered equivalent to those obtained in one mid-term examination. Zero marks will be awarded to students who are absent from the mid-term examination. The average of the best two scores from the three exams (the two mid-term exams and the CBT), combined with other internal assessment components, will constitute the Continuous Internal Improvement (CII) marks for that specific course.

9.4 Semester End Examination for theory courses

9.4.1 Theory Courses:

The semester end examinations (SEE), for theory courses, will be conducted for 60 marks consisting of two parts viz. i) Part- A for 10 marks and ii) Part - B for 50 marks.

- Part-A is compulsory, consists of ten short answer questions covering all units of syllabus; each question carries one marks.

- Part-B consists of five questions carrying 10 marks each. There shall be two questions asked in the question paper from each unit with either-or choice and the student should answer either of the two questions. The student shall answer one question from each of five units.

9.4.2 Graphics for Engineers Course:

Question paper consists of five questions carrying 12 marks each. There shall be two questions asked in the question paper from each unit with either-or choice and the student should answer either of the two questions. The student shall answer one question from each of five units. There shall be no section with short answer questions.

9.4.3 Duration of SEE:

The duration of Semester End Examination of theory and graphics for engineers courses is 3 hours.

9.5 Continuous Internal Evaluation and Semester End Examination for Practical Courses

For practical courses there shall be a Continuous Internal Evaluation (CIE) during the semester for 40 marks and semester-end examination for 60 marks. The breakup of the continuous internal evaluation for 40 marks is as follows:

1. 10 marks for a write-up on day-to-day experiments in the laboratory (in terms of aim, components/procedure, expected outcome).
2. 10 marks for viva-voce (or) tutorial (or) case study (or) application (or) poster presentation of the course concerned.
3. 10 marks for the internal practical examination conducted by the laboratory teacher concerned.
4. The remaining 10 marks are for G-Lab on Board (G-LOB)/Project and Presentation, which consists of the Design (or) Software / Hardware Model Presentation (or) App Development (or) Prototype submission which shall be evaluated after completion of laboratory course and before semester end practical examination.

The Semester End Examination for practical courses shall be conducted with an external examiner and the laboratory course teacher. The external examiner shall be appointed from the college outside their cluster and not from a group colleges.

In the Semester End Examination for practical courses held for 3 hours, rubrics of evaluation for 60 marks is as given below:

1. 10 marks for write-up
2. 15 for experiment/program

3. 15 for evaluation of results
4. 10 marks for presentation on another experiment/program in the same laboratory course and
5. 10 marks for viva-voce on concerned laboratory course.

For any change of experiment, 5 marks will be deducted from the total of 60 marks. If second time change of experiment is requested, another five marks will be deducted from the 60 marks. No third change will be permitted.

9.6 Field-based Research Project:

There shall be a Field-based Research Project in the intervening summer between II-II and III- I Semesters. Students will register for this project immediately after II Year II Semester examinations and pursue it during summer vacation. The Field-based Research Project shall be submitted in a report form and presented before the committee in III year I semester. It shall be evaluated for 100 external marks. The evaluation committee shall consist of an External Examiner, Head of the Department, Supervisor of the Project and a Senior Faculty Member of the department. There shall be no internal marks for Field-based Research Project. Student shall have to earn 40% marks, i.e 40 marks out of 100 marks. The student is deemed to have failed, if he (i) does not submit a report on the Project, or (ii) does not make a presentation of the same before the committee as per schedule, or (iii) secures less than 40% marks in this course.

9.7 Internship/Industry Oriented Mini Project:

There shall be an Internship/Industry Oriented Mini Project in collaboration with an industry from their specialization. Students shall register for this project immediately after III Year II Semester Examinations and pursue it during summer vacation. Internship should be carried out at an organization (or) Industry. The Industry Oriented Mini Project shall be submitted in a report form and presented before the committee in IV Year I Semester before the semester end examination. It shall be evaluated for 100 external marks. The committee consists of an External Examiner, Head of the Department, Supervisor of the Industry Oriented Mini Project/Internship, and a Senior Faculty Member of the Department.

9.7.1 For evaluating industry-oriented mini-projects, it is preferable to appoint an external examiner from the industry, ideally from one of the organizations/ industries with which the institute has established / proposing to establish collaborations.

9.8 UG Project Work:

9.8.1 The UG project work shall be initiated at the beginning of the IV Year II Semester and the duration of the project work is one semester. The student must present in consultation with his/her supervisor, the title, objective and plan of action of his/her Project work to the departmental committee for approval within two weeks from the commencement of IV Year II Semester. Only after obtaining the approval of the departmental committee, the student can start his/her project work.

9.8.2 Student has to submit project work report at the end of IV Year II Semester. The project work shall be evaluated for 100 marks. Out of which 40 marks and 60 marks are allocated for CIE and External Evaluation respectively.

9.8.3 For internal evaluation, the departmental committee consisting of Head of the Department, Project Supervisor and a Senior Faculty Member shall evaluate the project work for 40 marks. The distribution of marks is as follows:

• Objective(s) of the work done	- 05 Marks
• Methodology adopted	- 15 Marks

- Results and Discussions - 15 Marks
- Conclusions and Outcomes - 05 Marks
- Total - 40 Marks

9.8.4 The External Evaluation shall be conducted by the external examiner for a total of 60 marks. It shall comprise the presentation of the work, communication skills, and viva-voce, with a weightage of 20 marks, 15 marks, and 25 marks respectively.

The topics for main Project shall be different from the topic of Industry Oriented Mini Project/ Internship/SDC. The student is deemed to have failed, if he (i) does not submit a report on the Project, or (ii) does not make a presentation of the same before the External Examiner as per schedule, or (iii) secures less than 40% marks in the sum total of the CIE and SEE taken together.

9.8.5 For conducting viva-voce exam of project work, Controller of Examination appoints an external examiner. The external examiner may be selected from the list of experts submitted by the Head of the department.

9.8.6 A student who has failed, may re-appear once for the above evaluation, when it is scheduled again; if student fails in such ‘one re-appearance’ evaluation also, he/she has to appear for the same in the next subsequent year, as and when it is scheduled.

9.9 Skill Development Courses:

Skill Development Courses are included in the Curriculum. Each Skill Development Course carries one credit. The evaluation pattern will be same as that of a laboratory course including the internal and external assessments.

The objective of Skill Courses is to develop the cognitive skills as well as the psychomotor skills.

9.10 Value-Added Courses:

The evaluation of Value-Added Courses shall be similar to that of theory courses. However, the scheduling of these mid-term exams and semester-end examinations may not be combined with main-stream examinations. One hour /45 mins proctored mid-term examination shall be conducted in the regular class by the same subject teacher. It should not impact the conduct of other classes on that day. The scheduling of the semester-end examinations shall also be intimated by the controller of examination from time to time.

10. Grading Procedure

10.1 Absolute grading system is followed for awarding the grades to each course.

10.2 Grades will be awarded to indicate the performance of students in each Theory, Laboratory, Industry-Oriented Mini Project/ Internship/ Skill development course and Project Work. Based on the percentage of marks obtained (Continuous Internal Evaluation plus Semester End Examination, both taken together) as specified in clause 8 above, a letter grade shall be given as explained in the following clause.

10.3 To measure the performance of a student, a 10-point grading system is followed. The mapping between the percentage of marks secured and the corresponding letter grade is as follows:

Letter Grade	Grade Point	Percentage of marks
O (Outstanding)	10	Marks >= 90

A+ (Excellent)	9	Marks >= 80 and Marks < 90
A (Very Good)	8	Marks >= 70 and Marks < 80
B+ (Good)	7	Marks >= 60 and Marks < 70
B (Average)	6	Marks >= 50 and Marks < 60
C (Pass)	5	Marks >= 40 and Marks < 50
F (Fail)	0	Marks < 40
Ab (Absent)	0	Absent

Letter grade ‘F’ in any Course implies failure of the student in that course and no credits of the above table are earned.

10.4 Computation of SGPA and CGPA:

The UGC recommends the following procedure to compute the Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA):

i) **Sk** the SGPA of **k**th semester (1 to 8) is the ratio of sum of the product of the number of credits and grade points to the total credits of all courses registered by a student, i.e.,

$$\boldsymbol{GPA\ (S_k) = \sum_{i=1}^{n}(\ C_i * G_i) / \sum_{i=1}^{n} C_i}$$

Where **Ci** is the number of credits of the **i**th course and **Gi** is the grade point scored by the student in the **i**th course and **n** is the number of courses registered in that semester.

ii) The CGPA is calculated in the same manner taking into account all the courses **m**, registered by student over all the semesters of a programme, i.e., up to and inclusive of **Sk**, where **k ≥ 2**.

$$\boldsymbol{CGPA = \sum_{i=1}^{m}(C_i * G_i) /\ \sum_{i=1}^{m} C_i}$$

iii) The CGPA of the entire B.Tech programme shall be calculated considering the best 160 credits earned by the student.

iv) The SGPA and CGPA shall be rounded off to 2 decimal points.

11. Promotion Rules

S.No.	Promotion	Conditions to be Fulfilled
1	First year first semester to first year second semester	Regular course of study of first year first semester and fulfilment of attendance requirement.
2	First year second semester to Second year first semester	(i) Regular course of study of first year second semester and fulfilment of attendance requirement (ii) Must have secured at least **25%** of the total credits up to first year second semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or

		not.
3.	Second year first semester to Second year second semester	Regular course of study of second year first semester and fulfilment of attendance requirement.
4	Second year second semester to Third year first semester	(i) Regular course of study of second year second semester and fulfilment of attendance requirement. (ii) Must have secured at least **25%** of the total credits up to second year second semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or not.
5	Third year first semester to Third year second semester	Regular course of study of third year first semester and fulfilment of attendance requirement.
6	Third year second semester to Fourth year first semester	Regular course of study of third year second semester and fulfilment of attendance requirement.
7	Fourth year first semester to Fourth year second semester	Regular course of study of fourth year first semester and fulfilment of attendance requirement.

12. Re-admission after Detention

- A student detained due to lack of credits, shall be promoted to the next academic year only after acquiring the required number of credits.

- A student detained due to shortage of attendance shall be admitted in the same semester in the successive academic years.

- When a student is readmitted in the following academic years, the academic regulations under which the student seeks re-admission shall only be applicable to this student, not the academic regulations in which he got admitted in his/her first year of study.

13. Credit Exemption

A student (i) shall register for all courses covering 164 credits as specified and listed in the course structure and (ii) earn 160 or more credits to successfully complete the undergraduate programme.

- Best 160 credits shall be considered for CGPA computation. The student can avail exemption of courses totaling up to 4 credits other than Professional core courses, Laboratory Courses, Seminars, Project Work and Field Based Research Project / Industry Oriented Mini Project / Internship, for optional drop out from these 164 credits registered;

- The semester grade point average (SGPA) of each semester shall be mentioned at the bottom of the grade card, when all the subjects in that semester have been passed by the student.

- Credits earned by the student in either a Minor or Honors program cannot be counted towards the required 160 credits for the award of the B.Tech degree.

14. Award of Degree:

14.1 After a student satisfies all the requirements prescribed for the completion of the Degree and becomes eligible for the award of B.Tech Degree by JNTUH, he/she shall be placed in one of the following four classes based on CGPA secured from the 160 credits.

S. No	Class Awarded	CGPA Secured
1	First Class with Distinction	CGPA >= 7.50 with no F or below grade/ detention anytime during the programme
2	First Class	CGPA >= 7.50 with rest of the clauses of S.No 1 not satisfied
3	First Class	CGPA >= 6.50 and CGPA < 7.50
4	Second Class	CGPA >=5.50 and CGPA < 6.50
5	Pass Class	CGPA >= 5.00 and CGPA < 5.50

Equivalence of grade to marks
Marks % = (CGPA -0.5)*10

14.2 Grace Marks

Grace marks shall be given to those students who complete the course work of four year B.Tech degree, not secured pass grade in not more than three subjects and adding a specified grace marks enables the student to pass the subject(s) as well as gets eligibility to receive the provisional degree certificate.
Grace marks for students admitted under the GR25 Academic Regulations should not exceed 0.15% of the total maximum marks in all eight semesters (excluding the marks allocated for value added courses and skill development courses).

15. Multiple Entry Multiple Exit Scheme (MEME)

15.1 Exit Option after Second Year:

Students enrolled in the 4-Year B.Tech program are permitted to exit the program after successful completion of the second year (B.Tech II Year II Semester). The students who desire to exit after the II year shall formally inform the exit plan one semester in advance i.e. at the commencement of II Year II Semester itself. Such students need to fulfil the additional requirements as specified in Clause 15.2 described below.

Upon fulfilling the requirements like earning all the credits up to II Year II Semester and successfully completing the additional requirements, the students will be awarded a 2-Year Undergraduate (UG) Diploma in the concerned engineering branch.

15.2 Additional Requirements for Diploma Award

To qualify for the diploma under the exit option, students must also complete 2 additional credits through one of the following University-prescribed pathways:

Work-based Vocational Course:

Participation in a practical, hands-on vocational training program relevant to the engineering field, typically conducted during the summer term.

Internship/Apprenticeship:

Completion of a minimum 8-week internship or apprenticeship in their related field to gain practical industry exposure. In addition, students must clear any associated course(s) and submit the internship/ apprenticeship report.

15.3 Re-entry into the B.Tech Programme

Students who have exited the B.Tech program with a 2-Year UG Diploma may apply for re- entry into the Third Year (Fifth Semester) of the B.Tech program. Re-entry is subject to the following conditions:

- The student must surrender the awarded UG Diploma Certificate.
- Students who wish to rejoin in III Year must join the same B.Tech program and same college from which the student exited. Before rejoining, students should check for continuation of the same branch at the college. If the specific branch is closed in that particular college, then student should consult the University for the possible alternative solutions.
- Re-registered students will be governed by the academic regulations in effect at the time of re-entry, regardless of the original regulations under which they were admitted.
- If a student opts to continue his/her studies without a gap after being awarded the diploma, they must register for the third-year courses before the commencement of classwork.

15.4 Break in Study and Maximum Duration

Students are allowed to take a break of up to four years after completion of II Year II Semester with prior permission.

Re-entry after such a break is subject to the condition that the student completes all academic requirements within twice the duration of the program (i.e., within 8 years for a 4-year B.Tech programme).

16. Transitory Regulations for the students re-admitted in GR25 Regulations:

16.1 Transitory regulations are applicable to the students detained due to shortage of attendance as well as detained due to the shortage of credits and seek permission to re-join the B.Tech programme, where GR25 regulations are in force.

16.2 A student detained due to shortage of attendance and re-admitted in GR25 regulations: Such students shall be permitted to join the same semester, but in GR25 Regulations.

16.3 A student detained due to shortage of credits and re-admitted in GR25 regulations: Such students shall be promoted to the next semester in GR25 regulations, only after acquiring the required number of credits as per the corresponding regulations of his/her previous semester.

16.4 A student who has failed in any course in a specific regulation has to pass those courses in the same regulations.

16.5 If a student is readmitted to GR25 Regulations and has any course with 80% of syllabus common with his/her previous regulations, that particular course in GR25 Regulations will be substituted by an equivalent course of previous regulations

16.6 The GR25 Academic Regulations are applicable to a student from the year of re-admission. However, the student is required to complete the study of B.Tech degree within the stipulated period of eight academic years from the year of first admission.

17 Student Transfers

17.1 There shall be no branch transfers after the completion of admission process.

17.2 There shall be no transfers from one college to another within the constituent colleges and units of Jawaharlal Nehru Technological University Hyderabad.

17.3 The students seeking transfer to colleges affiliated to JNTUH from various other Universities/institutions is having back-logs at the previous University/institute, have to pass the courses offered at JNTUH which are equivalent to the failed courses at the previous University/institute.

17.4 The transferred students from other Universities/Institutions to JNTUH affiliated colleges, shall be given a chance to write CBTs for getting CIE component in the equivalent course(s) as per the clearance letter issued by the University.

18 Honors and Minor Degree Programmes

Honors Degree programme is available for B.Tech CSE and Minor Degree programme is available in Artificial Intelligence & Machine Learning for all branches of B.Tech. degree except for B.Tech CSE(AIML). Minor Degree programmes will commence from II Year II Semester and continue till IV Year I semester and Honors Degree programmes will commence from III Year I Semester and continue till IV Year II Semester.

Academic Regulations for B.Tech (Lateral Entry) under GR25

(Applicable for Batches Admitted from 2025-26)

1. All regulations as applicable for B.Tech 4-year degree programme (Regular) will hold good for B.Tech (Lateral Entry Scheme) except for the following rules:

a) Pursued programme of study for not less than three academic years and not more than six academic years.

b) A student should register 123/124 credits and secure 120 credits. The marks obtained in all 120 credits shall be considered for the calculation of the final CGPA.

c) The student can avail exemption of courses totaling up to 3/4 credits other than Professional core courses, Laboratory Courses, Seminars, Project Work and Field Based Research Project/ Industry Oriented Mini Project / Internship, for optional drop out.

d) Lateral Entry students are not permitted to exit the B.Tech. program after completion of second year (B.Tech. II Year II Semester).

e) Students who fail to fulfil all the academic requirements for the award of the degree within six academic years from the year of their admission, shall forfeit their seat in B.Tech programme.

2. Academic Requirements and Promotion Rules:

a) A student shall be deemed to have satisfied the minimum academic requirements and earned the credits allotted to each theory or laboratories if he/she secures not less than 35% of marks in the Semester-End Examination and a minimum of 40% of the sum total of the Internal Evaluation and Semester-End Examination taken together.

b) A student shall be promoted to the next year only when he/she satisfies the requirements of all the previous semesters.

S. No	Promotion	Conditions to be fulfilled
1	Second year first semester to Second year second semester	Regular course of study of second year first semester and fulfilment of attendance requirement.
2	Second year second semester to Third year first semester	(i) Regular course of study of second year second semester and fulfilment of attendance requirement. (ii) Must have secured at least **25%** of the total credits up to second year second semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or not.
3	Third year first semester to Third year second semester	Regular course of study of third year first semester and fulfilment of attendance requirement.
4	Third year second semester to Fourth year first semester	Regular course of study of third year second semester and fulfilment of attendance requirement.

5	Fourth year first semester to Fourth year second semester	Regular course of study of fourth year first semester and fulfilment of attendance requirement.

3. **Award of Class:** After a student satisfies all the requirements prescribed for the completion of the Degree and becomes eligible for the award of B.Tech Degree by JNTUH, he/she shall be placed in one of the following four classes based on CGPA secured from the 120 credits.

S. No	**Class Awarded**	**CGPA Secured**
1	First Class with Distinction	CGPA >= 7.50 with no F or below grade/ detention anytime during the Programme
2	First Class	CGPA >= 7.50 with rest of the clauses of S.no 1 not satisfied
3	First Class	CGPA >=6.50 and CGPA < 7.50
4	Second Class	CGPA >= 5.50 and CGPA < 6.50
5	Pass Class	CGPA >= 5.00 and CGPA < 5.50

Academic Regulations for B.Tech with Minors Programme under GR25

(Applicable for Batches Admitted from 2025-26)

1. Objectives

The key objectives of offering B.Tech with Minor programme are:

- To expand the domain knowledge of the students in one of the other programmes of engineering.
- To increase the employability of undergraduate students keeping in view of better opportunity in interdisciplinary areas of engineering & technology.
- To provide an opportunity to students to pursue their higher studies in the inter-disciplinary areas in addition to their own programme of study.
- To offer the knowledge in the areas which are identified as emerging technologies/thrust areas of Engineering.

2. Academic Regulations for B.Tech Degree with Minor programmes

a) The weekly instruction hours, internal & external evaluation and award of grades are on par with regular 4 -Years B.Tech programme.

b) For B.Tech with Minor, a student needs to earn additional 18 credits (over and above the required 160 credits for B.Tech degree). Minor Degree programmes will commence from II Year II Semester and continue till IV Year I Semester.

c) After registering for the Minor programme, if a student is unable to earn all the required 18 credits in a specified duration (twice the duration of the course), he/she shall not be awarded Minor degree. However, if the student earns all the required 160 credits of B.Tech, he/she will be awarded only B.Tech degree in the concerned programme.

d) There is no transfer of credits from Minor programme courses to regular B.Tech degree course and vice versa.

e) These 18 credits are to be earned from the additional Courses offered by the host department in the college as well as from the MOOCs platform.

f) For the course selected under MOOCs platform following guidelines may be followed:

i) Prior to registration of MOOCs courses, formal approval of the courses, by the University is essential. University before the issue of approval considers the parameters like the institute / agency which is offering the course, syllabus, credits, duration of the programme and mode of evaluation etc.

ii) Minimum credits for MOOCs course must be equal to or more than the credits specified in the Minor course structure provided by the University.

iii) Only Pass-grade/marks or above shall be considered for inclusion of grades in minor grade memo.

iv) Any expenses incurred for the MOOCs courses are to be met by the students only.

g) The option to take a Minor programme is purely the choice of the student.

h) The student shall be given a choice of withdrawing all the courses registered and/or the credits earned for Minor programme at any time; and in that case the student will be awarded only B.Tech degree in the concerned programme on earning the required credits of 160.

i) The student can choose only one Minor programme along with his/her basic engineering degree. A student who chooses an Honors programme is not eligible to choose a Minor programme and vice-versa.

j) A student can graduate with a Minor if he/she fulfils the requirements for his/her regular B.Tech programme as well as fulfils the requirements for Minor programme.

k) The institute shall maintain a record of students registered and pursuing their Minor programmes, minor programme-wise and parent programme -wise. The same report needs to be sent to the University once the enrolment process is complete.

l) The institute / department shall prepare the time-tables for each Minor course offered at their respective institutes without any overlap/clash with other courses of study in the respective semesters.

3. Eligibility conditions for the student to register for Minor programme

a) A student can opt for B.Tech programme with Minor programme if she/he has no active backlogs till II Year I Semester (III semester) at the time of entering into II year II semester.

b) Prior approval of mentor and Head of the Department for the enrolment into Minor programme, before commencement of II year II Semester (IV Semester), is mandatory

c) If more than 50% of the students in a programme fulfil the eligibility criteria (as stated above), the number of students given eligibility should be limited to 50%.

4. Registration for the courses in Minor Programme

a) At the beginning of each semester, just before the commencement of classes, students shall register for the courses which they wish to take in that semester.

b) The students should choose a course from the list against each semester (from Minors course structure) other than the courses they have studied/registered for regular B.Tech programme. No course should be identical to that of the regular B.Tech course. The students should take the advice of faculty mentors while registering for a course at the beginning of semester.

c) The maximum No. of courses for the Minor is limited to two (three in case of inclusion of lab) in a semester along with regular semester courses.

d) The registration fee to be collected from the students by the College is **Rs. 1000/-** per one credit.

e) A fee for late registration may be imposed as per the norms.

5. Minor courses and the offering departments

S. No.	Minor Programme	Eligible programme of students	@Offering Department	Award of Degree
1.	Artificial Intelligence & Machine Learning	All programmes, except B.Tech in CSE (AI&ML) /B.Tech (AI&ML)/ B.Tech (AI)/ B.Tech CSE(AI)	CSE	"B.Tech in programme name with Minor in Artificial Intelligence & Machine Learning"

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

Bachupally, Kukatpally, Hyderabad–500090, India. (040)65864440

COMPUTER SCIENCE AND BUSINESS SYSTEM (CSBS)

GR25 COURSE STRUCTURE

I B. Tech – CSBS - I Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	Maths	BS	GR25A1001	Linear Algebra and Function Approximation	3	1	0	4
2	Physics	BS	GR25A1003	Advanced Engineering Physics	3	0	0	3
3	ECE	ES	GR25A1030	Semiconductor Devices and Circuits	2	0	0	2
4	English	HS	GR25A1005	English for Skill Enhancement	3	0	0	3
5	CSE	ES	GR25A1006	Programming for Problem Solving	2	0	0	2
6	CSE	ES	GR25A1009	Basics of Computer Science and Engineering	1	0	0	1
7	ME	ES	GR25A1015	Graphics forEngineers	1	0	4	3
8	Physics	BS	GR25A1017	Advanced Engineering Physics Lab	0	0	2	1
9	CSE	ES	GR25A1020	Programming for Problem Solving Lab	0	0	3	1.5
10	English	HS	GR25A1019	English Language and Communication Skills Lab	0	0	2	1
		TOTAL			**15**	**1**	**11**	**21.5**

I B. Tech CSBS - II Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	Maths	BS	GR25A1002	Differential Equations and Vector Calculus	3	1	0	4
2	Chemistry	BS	GR25A1004	Engineering Chemistry	3	0	0	3
3	EEE	ES	GR25A1007	Fundamentals of Electrical Engineering	3	0	0	3
4	CSBS	ES	GR25A1028	Data Structures and Algorithms	2	0	0	2
5	CSE	ES	GR25A1026	Python Programming	1	0	0	1
6	Mgmt	HS	GR25A1027	Innovation and Design Thinking	1	0	0	1
7	ME	ES	GR25A1024	Engineering Workshop	1	0	3	2.5
8	Chemistry	BS	GR25A1018	Engineering Chemistry Lab	0	0	2	1
9	EEE	ES	GR25A1022	Fundamentals of Electrical Engineering Lab	0	0	2	1
10	CSBS	ES	GR25A1029	Data Structures and Algorithms Lab	0	0	2	1
		TOTAL			**14**	**1**	**9**	**19.5**

II B. Tech –CSBS – I Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSE	PC	GR25A2076	Discrete Mathematics	3	0	0	3
2	CSE(DS)	PC	GR25A2083	Computer Organization and Architecture	3	0	0	3
3	CSE	PC	GR25A2070	Java Programming	3	0	0	3
4	CSBS	PC	GR25A2084	Computational Statistics	3	0	0	3
5	CSE	PC	GR25A2071	Database Management Systems	3	0	0	3
6	Chemistry	VAC	GR25A2001	Environmental Science	1	0	0	1
7	CSE	PC	GR25A2072	Java Programming Lab	0	0	4	2
8	CSBS	PC	GR25A2085	Computational Statistics Lab	0	0	2	1
9	CSE	PC	GR25A2073	Database Management Systems Lab	0	0	3	1.5
10	CSE(DS)	SD	GR25A2097	Data Visualization Lab	0	0	3	1.5
TOTAL					**16**	**0**	**12**	**22**

II B. Tech –CSBS - II Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSBS	PC	GR25A2086	Operating Systems Concepts	3	0	0	3
2	CSBS	PC	GR25A2087	Principles of Software Engineering	3	0	0	3
3	CSBS	PC	GR25A2088	Algorithm Design and Analysis	3	0	0	3
4	ME	PC	GR25A2089	Operational Research	3	0	0	3
5	Maths	BS	GR25A2006	Mathematical And Statistical Foundations	3	0	0	3
6	English	VAC	GR25A2003	Indian Knowledge System	1	0	0	1
7	CSBS	PC	GR25A2090	Operating Systems Concepts Lab	0	0	3	1.5
8	CSBS	PC	GR25A2091	Principles of Software Engineering Lab	0	0	3	1.5
9	CSBS	PC	GR25A2092	Algorithm Design and Analysis Lab	0	0	3	1.5
TOTAL					**16**	**0**	**9**	**20.5**

III B. Tech CSBS – I Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSBS	PC		Web Technologies	3	0	0	3
2	Mgmt	HS		Fundamentals of Management	3	0	0	3
3	CSBS	PC		Machine Learning with R	3	0	0	3
4		PE		**Professional Elective I**	3	0	0	3
5	CSBS	OE		**Open Electives-I**	3	0	0	3
6	CSBS	PC		Web Technologies Lab	0	0	3	1.5
7	CSBS	PC		Machine Learning with R Lab	0	0	4	2
8	CSBS	PW		Field -based Research Project	0	0	4	2
TOTAL					**15**	**0**	**11**	**20.5**

PROFESSIONAL ELECTIVE – I								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSE	PE		Automata and Compiler Design	3	0	0	3
2	CSBS	PE		Cloud, Microservices and Application	3	0	0	3
3	CSE (DS)	PE		Big Data Analytics	2	1	0	3
4	CSBS	PE		Internet of Things	2	1	0	3

OPEN ELECTIVE – I								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSBS	OE		Software Design with UML	2	1	0	3
2	CSE(DS)	OE		R Programming	2	1	0	3
3	CE	OE		Geographic Information Systems and Science	3	0	0	3
4	ME	OE		Composite Materials	3	0	0	3
5	EEE	OE		Concept of Control Systems	3	0	0	3
6	ECE	OE		Principles of Communications	3	0	0	3
7	CSE(AIML)	OE		Basics of Java Programming	3	0	0	3
8	CSE	OE		Data Science for Engineers	3	0	0	3

III B. Tech CSBS – II Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSBS	PC		Computer Communications	3	0	0	3
2	CSBS	PC		Fundamentals of Artificial Intelligence	3	0	0	3
3	Mgmt	HS		Economics and Accounting for Engineers	3	0	0	3
4		PE		**Professional Elective II**	3	0	0	3
5	CSBS	OE		**Open Elective-II**	3	0	0	3
6	Mgmt	VAC		Values Ethics and Gender Culture	1	0	0	1
7	CSBS	PC		Linux Programming Lab	0	0	3	1.5
8	CSBS	PC		Fundamentals of Artificial Intelligence Lab	0	0	3	1.5
9	English	BS		Advanced English Communication Skills Lab	0	0	2	1
TOTAL					**16**	**0**	**8**	**20**

PROFESSIONAL ELECTIVE – II								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSBS	PE		Recommender Systems	2	1	0	3
2	CSE(DS)	PE		Data Science	2	1	0	3
3	CSE(AIML)	PE		Mobile Computing	3	0	0	3
4	Mgmt	PE		Enterprise Resource Planning	3	0	0	3

OPEN ELECTIVE – II								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSBS	OE		Business Analytics	2	1	0	3
2	CSE (DS)	OE		Web and Social Media Analytics	2	1	0	3
3	CE	OE		Plumbing-Water and Sanitation	3	0	0	3
4	ME	OE		Operations Research	3	0	0	3
5	EEE	OE		Artificial Neural Networks and Fuzzy Logic	3	0	0	3
6	ECE	OE		Hardware for IoT Applications	3	0	0	3
7	CSE (AIML)	OE		Introduction to DBMS	3	0	0	3
8	CSE	OE		Data Analytics	3	0	0	3

IV B. Tech –CSBS – I Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSE(AIML)	PC		Deep Learning	3	0	0	3
2	Mgmt	HS		Human Resource Management	3	0	0	3
3	CSBS	PE		**Professional Elective III**	2	1	0	3
4	CSBS	PE		**Professional Elective IV**	3	0	0	3
5		**OE**		**Open Elective- III**	3	0	0	3
6	CSBS	PC		DevOps Lab	0	0	2	1
7	CSBS	PC		Deep Learning Lab	0	0	4	2
8	CSBS	PW		Industry Oriented Mini Project/Internship	0	0	4	2
				TOTAL	**14**	**1**	**10**	**20**

PROFESSIONAL ELECTIVE – III								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSBS	PE		Cognitive Science and Analytics	2	1	0	3
2	CSBS	PE		Software Project Management	2	1	0	3
3	CSE(DS)	PE		Information Retrieval Systems	2	1	0	3
4	CSE(DS)	PE		Game Theory	2	1	0	3

PROFESSIONAL ELECTIVE – IV								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	Mgmt	PE		Marketing Management and Research	3	0	0	3
2	CSE(AIML)	PE		Generative AI	3	0	0	3
3	CSE	PE		Natural Language Processing	3	0	0	3
4	CSBS	PE		Information System Audit and Assurance	3	0	0	3

OPEN ELECTIVE – III

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1	CSBS	OE		E-Commerce	3	0	0	3
2	CSE (DS)	OE		Web and Social Media Analytics	2	1	0	3
3	CE	OE		Plumbing (Water and Sanitation)	3	0	0	3
4	ME	OE		Operations Research	3	0	0	3
5	EEE	OE		Artificial Neural Networks and Fuzzy Logic	3	0	0	3
6	ECE	OE		Hardware for IoT Applications	3	0	0	3
7	CSE (AIML)	OE		Introduction to Data Mining	3	0	0	3
8	CSE	OE		Augmented Reality and Virtual Reality	3	0	0	3

IV B. Tech –CSBS - II Semester

S.No	BOS	Group	Course Code	Course Name	L	T	P	Credits
1		PE		**Professional Elective V**	3	0	0	3
2		PE		**Professional Elective VI**	3	0	0	3
3	CSBS	PW		Project Work	0	0	42	14
TOTAL					**6**	**0**	**42**	**20**

PROFESSIONAL ELECTIVE – V								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	Mgmt	PE		Behavioral Economics	3	0	0	3
2	Mgmt	PE		Computational Finance & Modeling	3	0	0	3
3	CSBS	PE		Robotic Process Automation	2	1	0	3
4	CSBS	PE		IT Project Management	2	1	0	3

PROFESSIONAL ELECTIVE – VI								
S.No	**BOS**	**Group**	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**
1	CSBS	PE		Green Computing	3	0	0	3
2	CSBS	PE		Business Intelligence	2	1	0	3
3	CSE	PE		Cyber Security	3	0	0	3
4	Mgmt	PE		Financial Management	3	0	0	3

PROFESSIONAL ELECTIVES – 4 THREADS

S. No.	Theory and Algorithms	Applications	Data Science and Machine Intelligence	Software and Technology
1	Automata and Compiler Design	Cloud, Microservices and Applications	Big Data Analytics	Internet of Things
2	Recommender Systems	Mobile Computing	Data Science	Enterprise Resource Planning
3	Game Theory	Information Retrieval Systems	Cognitive Science and Analytics	Software Project Management
4	Natural Language Processing	Marketing Management and Research	Generative AI	Information System Audit and Assurance
5	Computational Finance & Modeling	Behavioral Economics	Robotic Process Automation	IT Project Management
6	Cyber Security	Financial Management	Business Intelligence	Green Computing

OPEN ELECTIVES FOR GR25 REGULATIONS

THREAD 1	THREAD 2	OFFERED BY
1. Soft Skills and Interpersonal Communication 2. Human Resource Development and Organizational Behavior 3. Cyber Law and Ethics 4. Economic Policies in India 5. Constitution of India	Data Science for Engineers	CSE
	Data Analytics	
	Augmented Reality and Virtual Reality	
	Basics of Java Programming	CSE AIML)
	Introduction to DBMS	
	Introduction to Data Mining	
	Introduction to Internet of Things	CSE (DS)
	R Programming	
	Web and Social Media Analytics	
	Software Design with UML	CSBS
	Business Analytics	
	E-Commerce	
	Non-Conventional Energy Sources	EEE
	Concepts of Control Systems	
	Artificial Neural Networks and Fuzzy Logic	
	Digital Electronics for Engineers	ECE
	Automotive Electronics	
	Hardware for IoT Applications	
	Industrial Automation and Control	ME
	Composite Materials	
	Operations Research and Management	
	Engineering Materials for Sustainability	CE
	Geographic Information Systems and Science	
	Plumbing -Water and Sanitation	

I YEAR
I SEMESTER

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

LINEAR ALGEBRA AND FUNCTION APPROXIMATION

Course Code: GR25A1001 **L/T/P/C: 3/1/0/4**
I Year I Semester

Pre-requisites: Mathematical Knowledge at pre-university level

Course Outcomes: After learning the contents of this course, the student must be able to

1. Recognize Rank of the matrix and write the matrix representation of a set of linear equations and to analyze the solution of the linear system of equations.
2. Discovery the Eigen values and Eigen vectors, Reduce the quadratic form to canonical form using orthogonal transformations.
3. Identify the geometrical interpretation of mean value theorems and discovery points in an interval that satisfy the mean value theorem for a given function.
4. Estimate the extreme values of functions of two variables with/ without constraints.
5. Evaluate the multiple integrals and apply the concept to find areas, volumes.

UNIT I : Matrices

Operations on vectors and matrices - Vector norms- Rank of a matrix by Echelon form – Linear dependence and independence of vectors. System of linear equations : Solution of a linear algebraic system of equations (homogeneous and non-homogeneous) using Gauss elimination .

UNIT II : Eigen values and Eigen vectors

Eigen values – Eigen vectors and their properties – Diagonalization of a matrix – Orthogonal diagonalization of a symmetric matrix- Definiteness of a symmetric matrix.

Quadratic forms and Nature of the Quadratic Forms – Reduction of Quadratic form to canonical form by Orthogonal Transformation.

UNIT III: Single Variable Calculus

Mean value theorems : Rolle’s theorem – Lagrange’s Mean value theorem with their Geometrical Interpretation and applications – Cauchy’s Mean value Theorem – Taylor’s Series (All the theorems without proof). Approximation off a function by Taylor’s series

UNIT IV: Multivariable Calculus (Partial Differentiation and applications)

Partial Differentiation **:** Total derivative – Jacobian – Functional dependence & independence. Applications : Maxima and minima of functions of two variables and three variables using method of Lagrange multipliers.

Curve Tracing: Curve tracing in cartesian coordinates

UNIT V : Multivariable Calculus (Integration)

Evaluation of Double Integrals (Cartesian and polar coordinates) – change of order of integration (only Cartesian form) – Change of variables for double integrals (Cartesian to polar). Evaluation of Triple Integrals – Change of variables for triple integrals (Cartesian to Spherical and Cylindrical polar coordinates). Applications: Areas by double integrals and volumes by triple integrals.

TEXTBOOKS :

1. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. R.K. Jain and S.R.K. Iyengar, Advanced Engineering Mathematics, Narosa Publications, 5th Editon, 2016.

REFERENCE BOOKS :

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9thEdition, Pearson, Reprint, 2002.
3. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
4. H. K. Dass and Er. Rajnish Verma, Higher Engineering Mathematics, S Chand and Company Limited, New Delhi.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ADVANCED ENGINEERING PHYSICS

Course Code: GR25A1003 **L/T/P/C:3/0/0/3**
I Year I Semester

Course Outcomes:

1. Apply quantum mechanical principles to explain particle behaviour and energy band formation in solids.
2. Comprehend the characteristics of semiconductor devices and characterization of nanomaterials.
3. Classify magnetic and dielectric materials based on their properties for various applications.
4. Analyze the principles of Laser and fibre optics and their applications.
5. Outline quantum computing concepts and use of quantum gates.

UNIT - I: Quantum Mechanics

Principles of Quantum Mechanics: Introduction, de-Broglie hypothesis, Heisenberg uncertainty principle, physical significance of wave function, postulates of quantum mechanics: operators in quantum mechanics, eigen values and eigen functions, Schrödinger's time independent wave equation, particle in a 1D box.
Band Theory of Solids: Blochs theorem (qualitative), Kronig-Penney model (qualitative): E-k diagram, effective mass of electron, discrete energy levels, formation of energy bands, classification of solids into metals, semiconductors and insulators.

UNIT - II: Semiconductors & Nanomaterials

Semiconductors: Intrinsic and extrinsic semiconductors(qualitative), Variation of Fermi level with temperature and doping(qualitative), Hall Effect and its applications, direct and indirect band gap semiconductors, Construction and principle of operation of p-n junction diode, I-V characteristics of p-n junction diode and Zener diode. Principle, Construction, Working, Characteristics and Applications: LED and Solar cell.
Nanomaterials: Introduction, quantum confinement in nanomaterials, Surface to volume ratio, Synthesis methods: Top-Down Technique: Ball milling method, Bottom-Up technique: Sol-Gel method, X-ray diffraction: Bragg's law, calculation of average crystallite size using Debye Scherrer's formula, scanning electron microscopy (SEM): block diagram, working principle.

UNIT - III: Magnetic and Dielectric Materials

Magnetic materials: Introduction to magnetic materials, origin of magnetic moment - classification of magnetic materials – Dia, Para, Ferro, Weiss domain theory of ferromagnetism, hysteresis curve based on domain theory of ferromagnetism, soft and hard magnetic materials, applications: magnetic hyperthermia for cancer treatment, magnets for EV.
Dielectric material: Introduction to dielectric materials, types of polarization: electronics, ionic & orientation(qualitative), derivation of electronic and ionic polarizability; ferroelectric, piezoelectric, pyroelectric materials and their applications: Ferroelectric Random-Access Memory (Fe-RAM), load cell and fire sensor.

UNIT - IV: Laser and Fibre Optics

Lasers: Introduction to laser, Radiative transition: Absorption, Spontaneous and Stimulated emissions, characteristics of laser, Einstein coefficients and their relations, metastable state, population inversion, pumping, lasing action, Ruby laser, He-Ne laser, semiconductor diode laser, applications: Bar code scanner, LIDAR for autonomous vehicle.

Fiber Optics: Introduction to fibre optics, total internal reflection, construction of optical fibre, acceptance angle, numerical aperture, classification of optical fibres, losses in optical fibre, applications: optical fibre for
communication system, sensor for structural health monitoring.

UNIT - V: Quantum Computing

Introduction, linear algebra for quantum computation, Dirac's Bra and Ket notation and their properties, Hilbert space, Bloch's sphere, concept of quantum computer, classical bits, Qubits, multiple Qubit system, entanglement, quantum gates (Pauli's X,Y,Z gate, Hadamard gate), quantum computing system for information processing, evolution of quantum systems, challenges and advantages of quantum computing over classical computation.

TEXTBOOKS:

1. Charles Kittel, Introduction to Solid State Physics, John Wiley & Sons, Inc.
2. Thomas G. Wong, Introduction to Classical and Quantum Computing, Rooted Grove
3. Engineering Physics, B.K. Pandey, S. Chaturvedi – Cengage Learing
4. A textbook of Engineering Physics, Dr. M. N. Avadhanulu, Dr. P.G. Kshirsagar - S. Chand.

REFERENCE BOOKS:

1. Jozef Gruska, Quantum Computing, McGraw Hill
2. Michael A. Nielsen & Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.
3. John M. Senior, Optical Fiber Communications Principles and Practice, Pearson Education Limited.
4. Fundamentals of Semiconductor Devices, Second Edition, Anderson and Anderson, McGraw Hill.

Useful Links

- https://shijuinpallotti.wordpress.com/wp-content/uploads/2019/07/optical-fibercommunications-principles-and-pr.pdf
- https://dpbck.ac.in/wp-content/uploads/2022/10/Introduction-to-Solid-State-PhysicsCharles-Kittel.pdf
- https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf
- https://www.fi.muni.cz/usr/gruska/qbook1.pdf
- https://profmcruz.wordpress.com/wp-content/uploads/2017/08/quantum-computation-andquantum-information-nielsen-chuang.pdf

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

SEMICONDUCTOR DEVICES AND CIRCUITS

Course Code: GR25A1030 **L/T/P/C: 2/0/0/2**
I Year I Semester

Course Outcomes: By the end of this course, students will be able to:

1. Analyze the electrical characteristics and models of semiconductor diodes and apply them in rectifier and clipping circuits.
2. Evaluate the operation and configurations of Bipolar Junction Transistors (BJTs) and analyze their input and output characteristics.
3. Design appropriate biasing networks for BJTs and determine the operating point for amplifier applications.
4. Analyze transistor amplifier circuits using h-parameter models and assess performance for various configurations.
5. Analyze the structure, working, and characteristics of JFETs, MOSFETs, and advanced devices like FinFETs and CNTFETs, and compare modern device technologies.

UNIT - I:
Diode Characteristics and Applications: PN junction diode – I-V characteristics, Diode resistance and capacitance, Diode models (Ideal, Simplified, Piecewise Linear), Rectifiers – Half-wave, Full-wave(Center-tap and bridge), Capacitor filter for rectifiers, Clippers and clampers, Zener diode – I-Characteristics and voltage regulation.

UNIT - II:
Bipolar Junction Transistor (BJT): Structure and working principle of BJT, Current components and transistor action, Configurations: Common Base (CB), Common Emitter (CE), Common Collector (CC), Input and output characteristics, Determination of h-parameters from transistor characteristics.

UNIT - III:
BJT Biasing: Need for biasing and stabilization, Load line and operating point, Biasing techniques: Fixed bias, Collector-to-base bias, Voltage divider bias, Stability factors and thermal runaway

UNIT - IV:
Transistor Amplifiers: Transistor as a small-signal amplifier, h-parameter equivalent circuit, CE, CB,CC amplifier analysis using h-parameters, Approximate CE model – with and without emitter bypass capacitor.

UNIT - V:
Field Effect Transistors and JFET: Structure, operation, and characteristics, MOSFET: Enhancement and Depletion modes – Structure, operation, and characteristics.

TEXTBOOKS:
1. Millman, Jacob, and Christos C. Halkias. Electronic Devices and Circuits. Tata McGraw-

Hill,1991.
2. Boylestad, Robert L., and Louis Nashelsky. Electronic Devices and Circuit Theory. Pearson,11th ed., 2013.
3. Sedra, Adel S., and Kenneth C. Smith. Microelectronic Circuits. Oxford University Press, 7th ed., 2014.

REFERENCE BOOKS:

1. Bell, David A. Electronic Devices and Circuits. Oxford University Press, 5th ed., 2008.
2. Neamen, Donald A. Electronic Circuit Analysis and Design. McGraw-Hill, 2nd ed., 2001.
3. Salivahanan, S., and N. Suresh Kumar. Electronic Devices and Circuits. McGraw-Hill Education, 4th ed., 2017.
4. Razavi, Behzad. Fundamentals of Microelectronics. Wiley, 2nd ed., 2013.
5. Taur, Yuan, and Tak H. Ning. Fundamentals of Modern VLSI Devices. Cambridge University Press, 2nd ed., 2009.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGLISH FOR SKILL ENHANCEMENT

Course Code: GR25A1005 **L/T/P/C: 3/0/0/3**

I Year I Semester

Course Outcomes: Students will be able to

1. Use English Language effectively in spoken and written forms.
2. Comprehend the given texts and respond appropriately.
3. Communicate confidently in various contexts and different cultures.
4. Acquire proficiency in English including reading and listening comprehension, writing and speaking skills.
5. Convey complex ideas clearly and accurately in academic and professional settings

UNIT –I

Theme: **Perspectives**
Lesson on 'The Generation Gap' by Benjamin M. Spock from the prescribed textbook titled *English for the Young in the Digital World* published by Orient Black Swan Pvt. Ltd.

Vocabulary: The Concept of Word Formation -The Use of Prefixes and Suffixes - Words Often Misspelt - Synonyms and Antonyms

Grammar: Identifying Common Errors in Writing with Reference to Parts of Speech particularly Articles and Prepositions – Degrees of Comparison

Reading: Reading and Its Importance- Sub Skills of Reading – Skimming and Scanning.

Writing: Sentence Structures and Types -Use of Phrases and Clauses in Sentences- Importance of Proper Punctuation- Techniques for Writing Precisely – Nature and Style of Formal Writing.

UNIT –II

Theme: **Digital Transformation**
Lesson on '*Emerging Technologies*' from the prescribed textbook titled *English for the Young in the Digital World* published by Orient BlackSwan Pvt. Ltd.

Vocabulary: Homophones, Homonyms and Homographs

Grammar: Identifying Common Errors in Writing with Reference to Noun-pronoun Agreement and Subject-verb Agreement.

Reading: Reading Strategies-Guessing Meaning from Context – Identifying Main Ideas – Exercises for Practice

Writing: Paragraph Writing – Types, Structures and Features of a Paragraph - Creating Coherence – Linkers and Connectives - Organizing Principles in a Paragraph – Defining- Describing People, Objects, Places and Events – Classifying- Providing Examples or Evidence - Essay Writing - Writing Introduction and Conclusion.

UNIT –III

Theme: **Attitude and Gratitude**

Poems on '*Leisure'* by William Henry Davies and '*Be Thankful'* - Unknown Author from the prescribed textbook titled *English for the Young in the Digital World* published by Orient BlackSwan Pvt. Ltd.

Vocabulary: Words Often Confused - Words from Foreign Languages and their Use in English.

Grammar: Identifying Common Errors in Writing with Reference to Misplaced Modifiers and Tenses.

Reading: Sub-Skills of Reading – Identifying Topic Sentence and Providing Supporting Ideas - Exercises for Practice.

Writing: Format of a Formal Letter-Writing Formal Letters E.g.., Letter of Complaint, Letter of Requisition, Job Application with CV/Resume – Difference between Writing a Letter and an Email - Email Etiquette.

UNIT –IV

Theme: **Entrepreneurship**

Lesson on '*Why a Start-Up Needs to Find its Customers First*' by Pranav Jain from the prescribed textbook titled *English for the Young in the Digital World* published by Orient BlackSwan Pvt. Ltd.

Vocabulary: Standard Abbreviations in English – Inferring Meanings of Words through Context – Phrasal Verbs – Idioms.

Grammar: Redundancies and Clichés in Written Communication – Converting Passive to Active Voice and Vice-Versa.

Reading: Prompt Engineering Techniques– Comprehending and Generating Appropriate Prompts - Exercises for Practice

Writing: Writing Practices- Note Making-Précis Writing.

UNIT –V

Theme: **Integrity and Professionalism**

Lesson on '*Professional Ethics*' from the prescribed textbook titled *English for the Young in the Digital World* published by Orient BlackSwan Pvt. Ltd.

Vocabulary: Technical Vocabulary and their Usage– One Word Substitutes – Collocations.

Grammar: Direct and Indirect Speech **-** Common Errors in English (Covering all the other aspects of grammar which were not covered in the previous units)

Reading: Survey, Question, Read, Recite and Review (SQ3R Method) – Inferring the Meaning and Evaluating a Text- Exercises for Practice

Writing: Report Writing - Technical Reports- Introduction – Characteristics of a Report – Categories of Reports Formats- Structure of Reports (Manuscript Format) -Types of Reports - Writing a Technical Report.

Note: *Listening and Speaking skills which are given under Unit-6 in AICTE Model Curriculum are covered in the syllabus of ELCS Lab Course.*

- (**Note**: As the syllabus of English given in AICTE *Model Curriculum-2018 for B.Tech. First Year is* ***Open-ended***, besides following the prescribed textbook, it is required to prepare teaching/learning materials **by the teachers collectively** in the form of handouts based on the needs of the students in their respective colleges for effective teaching/learning in the class.)

TEXTBOOKS:

1. Board of Editors. *2025. English for the Young in the Digital World.* Orient Black Swan Pvt. Ltd.

REFERENCE BOOKS:

1. Swan, Michael. (2016). *Practical English Usage.* Oxford University Press. New Edition.
2. *Karal, Rajeevan. 2023. English Grammar Just for You.* Oxford University Press. New Delhi
3. 2024. *Empowering with Language: Communicative English for Undergraduates.* Cengage Learning India Pvt. Ltd. New Delhi
4. Sanjay Kumar & Pushp Lata. 2022. *Communication Skills – A Workbook.* Oxford University Press. New Delhi
5. Wood,F.T. (2007). *Remedial English Grammar.* Macmillan.
6. Vishwamohan, Aysha. (2013). *English for Technical Communication for Engineering Students.*
   Mc Graw-Hill Education India Pvt. Ltd.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

PROGRAMMING FOR PROBLEM SOLVING

Course Code: GR25A1006 **L/T/P/C: 2/0/0/2**

I Year I Semester

Course Outcomes:

1. Design algorithms and flowcharts for problem solving and apply the basic elements of C programming to solve simple computational problems.
2. Illustrate decision-making control structures and use functions, including recursion, to develop modular C programs.
3. Discover the need for arrays, searching, sorting, and strings in problem-solving and apply them.
4. Summarize pointer operations and implement structures and unions to solve programming problems.
5. Demonstrate file handling mechanisms, preprocessor directives, and command line arguments in C.

UNIT I - Introduction to Programming

Introduction to Algorithms: Representation of Algorithm, Flowchart, Pseudo code with examples, compiling and executing programs, syntax, and logical errors.

Introduction to C Programming Language: General Form of a C Program, C Language Elements, operators, precedence and associativity, expression evaluation, implicit and explicit type conversion, Formatting Numbers in Program Output.

UNIT II -Decision Making and Functions

Branching and Loops: Conditional branching with simple if, if-else, nested if-else, else if ladder, switch-case, loops: for, while, do-while, jumping statements: goto, break, continue, exit.

Functions: Top-Down Design and Structure Charts, function declaration, signature of a function, parameters and return type of a function, categories of functions, parameter passing techniques, passing arrays and strings to functions, recursion, merits and demerits of recursive functions, Scope of Names.

UNIT III -Arrays and Strings

Arrays: One and two-dimensional arrays, creating, accessing, and manipulating elements of arrays.

Searching and sorting: Introduction, Linear search, and Binary search. Bubble Sort, Insertion Sort, Selection Sort.

Strings: Introduction to strings, operations on characters, basic string functions available in C - strlen, strcat, strcpy, strrev, strcmp, String operations without string handling functions, arrays of strings.

UNIT IV - Pointers and Structures

Pointers: Pointers and the Indirection Operator, declaration and initialization of pointers, pointer to pointer, void pointer, null pointer, pointers to arrays, function pointer.

Structures and Unions: Defining structures, declaring and initializing structures, arrays within structures, arrays of structures, nested structures, pointers to structures, passing structures to functions, unions, and typedef.

UNIT V -File Handling and Preprocessor in C

Files: Text and binary files, creating, reading, and writing text and binary files, random access to files, and error handling in files.

Preprocessor: Commonly used preprocessor commands like include, define, undef, if, ifdef, ifndef, elif, command line arguments and enumeration data type.

Teaching methodologies:

- PowerPoint Presentations
- Tutorial Sheets
- Assignments

Textbooks:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. B.A. Forouzan and R.F. Gilberg, C Programming and Data Structures, Cengage Learning, 3rd Edition.

Reference Books:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.
2. R.G. Dromey, How to solve it by Computer, Pearson, 16th Impression.
3. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
4. Herbert Schildt, C: The Complete Reference, McGraw-Hill, 4th Edition.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

BASICS OF COMPUTER SCIENCE AND ENGINEERING

Course Code: GR25A1009 **L/T/P/C: 1/0/0/1**

I Year I Semester

Course Outcomes:

1. Interpret the working principles of functional units of a basic Computer.
2. Analyze steps in program development and types of operating systems.
3. Identify the significance of database systems and computer networks.
4. Develop applications using MS-Word and MS-Excel.
5. Design presentations using MS-PowerPoint and develop web pages using web designing tools.

UNIT – I

Basics of a Computer – Characteristics of computer, Generations, classification, Hardware– Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software– systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – Waterfall model, Agile, Types of computer languages – Programming, markup, scripting, Program Development – steps in program development, flowcharts, algorithms.

Operating systems: Functions of operating systems, types of operating systems, Examples of OS- MS-DOS, Windows, Linux, Installation and formatting of Windows OS.

UNIT – III

Database Management Systems: Database Vs File System, Database applications, types of DBMS, Database users, SQL – Types of SQL commands.

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, 5G communications – evolution, key technologies.

UNIT – IV

MS-Word: Introduction, MS-Word screen and its components- Office button—New, open, save, save as, print, close, Ribbon—Home, Insert, Page layout, References, Review, View. Example Applications - Resume preparation, Magazine Cover, Mail merge.

MS-Excel: Basics of Spreadsheet, MS-Excel screen and its components, Office button, Ribbon-Home, Insert, Page Layout, Formulas, Data, Review, View. Example Application- Employee Salary calculation.

UNIT – V

MS-PowerPoint: MS-PowerPoint screen and its components, Office button, Ribbon-Home, Insert, Design, Animations, Slideshow, Review, View. Example - Design a “Happy Birthday” card.

World Wide Web: Basics, role of HTML, CSS, XML, Tools for web designing, Social media.

Teaching methodologies:

- Power Point Presentations
- Tutorial Sheets
- Assignments

TEXTBOOKS:

1. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
2. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
2. Elements of computer science, Cengage.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

GRAPHICS OF ENGINEERS

Course Code: GR25A1015 **L/T/P/C: 1/0/4/3**

I Year I Semester

Prerequisites: Mathematics, Physics.

Course Outcomes:

1. Generate two dimensional drawings and curves by using AutoCAD commands.
2. Interpret projection methods and draw projections of a line or point objects located in different positions.
3. Imagine and generate multi-view projections of planes and solid objects located in different positions
4. Construct and interpret sectional views of an object and develop its solid surfaces.
5. Create isometric drawings from given orthographic views and familiar with isometric to orthographic transformations.

UNIT I

Introduction to AutoCAD software: user interface, tool bar -draw, modify, dimension, layers, setting drawing area, status bar, print setup, generation of two-dimensional drawings.
Engineering curves- Conic sections – ellipse, parabola and hyperbola- general method only; Cycloidal curves- Cycloid, epi-cycloid and Hypocycloid.

UNIT II

Orthographic projection – Introduction, definition, and classification of projections; pictorial and multi-view, significance of first and third angle methods of projections;
Projections of points -a point situated in the first, second, third and fourth quadrants.
Projections of straight lines – Line inclined to one reference plane and parallel to the other.

UNIT III

Projections of planes - definition and types of regular plane figures like triangle, square, pentagon, hexagon, and circle; projections of planes -a plane inclined to one reference plane and perpendicular to the other.
Projections of solids - definition and types of right regular solids objects like prism, cylinder, pyramid, and cone; Projections of Solids -with an axis inclined to one reference plane and parallel to the other.

UNIT IV

Sections of solids- Section and sectional views of regular solids- Prisms, Cylinders, Pyramids and Cone – concept of Auxiliary Views.
Development of surfaces- Development of lateral surfaces of right regular solids - Prisms, Pyramids, Cylinders and Cone.

UNIT V

Isometric views– isomeric views of lines, planes (polygons) and solids (Prisms, Cylinders, Pyramids, and Cone); compound solids, generation of Isometric line diagrams. Introduction to World Coordinate System and User Coordinate System.

Conversion of views - Isometric Views to Orthographic Views and Vice-versa, and Conventions.

TEXTBOOKS:

1. Engineering Drawing by N. D. Bhatt, Fiftieth Revised and Enlarged Edition:2011, Charotar Publishing House Pvt. Ltd.
2. Engineering Graphics by Basant Agrawal and C M Agrawal, fifth re-print 2011, Tata McGraw Hill Education Private Limited, New Delhi.

REFERENCE BOOKS:

1. Engineering Graphics with AutoCAD 2020 by James D. Bethune, Copyright © 2020 by Pearson Education, Inc. All rights reserved.
2. Engineering Graphics by M. B. Shah, B. C. Rana, S. N. Varma, Copyright © 2011 Dorling Kindersley (India) Pvt. Ltd, Licensees of Pearson Education in South Asia.
3. Engineering Drawing and Graphics by K Venu Gopal /New Age International Pvt. Ltd, Publishers, fifth edition, 2002.
4. Engineering Graphics by Koushik Kumar, Apurba Kumar Roy, Chikesh Ranjan, S Chand and Company limited, first edition 2019.
5. Engineering Drawing with Auto Cad by B. V. R. Gupta, M. Raja Roy, IK International Pub., 2009.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ADVANCED ENGINEERING PHYSICS LAB

Course Code: GR25A1017 **L/T/P/C: 0/0/2/1**

I Year I Semester

Course Outcomes:

1. Categorize semiconductors using Hall effect and energy gap measurement techniques.
2. Illustrate working of optoelectronic devices through experimental study.
3. Analyze the behavior of magnetic fields and their applications.
4. Infer the characteristics of Lasers and study of losses in optical fibers.
5. Determine the frequency of tuning fork through the phenomena of resonance.

List of Experiments:

1. Determination of energy gap of a semiconductor.
2. Determination of Hall coefficient and carrier concertation of a given semiconductor.
3. Study of V-I characteristics of pn junction diode.
4. Study of V-I characteristics of light emitting diode.
5. Study of V-I Characteristics of solar cell.
6. Determination of magnetic field along the axis of a current carrying coil.
7. a) Determination of wavelength of a laser using diffraction grating.

   b) Study of V-I & L-I characteristics of a given laser diode.
8. Determination of numerical aperture of a given optical fibre.
9. Determination of bending losses of a given optical fibre.
10. Determination of frequency of a tuning fork using Melde's arrangement.

Note: Any 8 experiments are to be performed.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

PROGRAMMING FOR PROBLEM SOLVING LAB

Course Code: GR25A1020 **L/T/P/C: 0/0/3/1.5**

I Year I Semester

Course Outcomes:

1. Develop C programs from algorithms using C elements, selection constructs, and test and debug them for correctness.
2. Employ loops and functions effectively to design modular solutions for computational problems.
3. Utilize arrays and strings to organize, manipulate, and process data in problem-solving contexts.
4. Apply searching and sorting methods and structure-based representations to manage and process data efficiently.
5. Demonstrate the use of pointers, and apply file handling along with preprocessor directives to enhance C program execution and management.

TASK 1

a. Write the program for the simple, compound interest.
b. Write a C program to implement relational, logical and bitwise operators.
c. Write a C program for finding the maximum , minimum of three numbers.
d. Write a C program to Convert Celsius temperature to Fahrenheit and vice versa using type conversion.

TASK 2

a. Write a C program to find the roots of a quadratic equation using if-else.
b. Write a C program to check the triangle type based on sides using nested if- else. (Equilateral, Isosceles, Scalene, invalid).
c. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement).

TASK 3

a. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
b. Write a C program check whether a given number is Armstrong number or not.
c. Write a C program check whether a given number is Strong number or not.

TASK 4

a. Write a program to display prime numbers between X to Y.
b. Write a C program to calculate the sum of following series:
   (i) $S1= 1+x/1+x^2/2+ x^3/3\ldots.+x^n/n$
   (ii) $S2= 1+x/1!-x^2/2!+ x^3/3!\ldots.+x^n/n!$

TASK 5

a. Write a C program to display the following patterns:

```
i)                                   ii)
          1                          1
       2    3                        2 2
    4     5     6                    3 3 3
 7       8     9      10             4 4 4 4
```

b. Write a C program to display the following patterns:

```
i)                       ii)
 $                          E
 $$$                       ED
 $$$$$                    EDC
 $$$                     EDCB
 $                      EDCBA
```

TASK 6

a. Write a C program to swap two numbers using parameter passing techniques.

b. Write a C program to implement factorial of a given integer using recursive and non-recursive functions.

c. Write a C program to print first ‘n’ terms of Fibonacci series using recursive and non-recursive functions.

TASK 7

a. Write a C program to find the minimum, maximum and average in an array of integers.

b. Write a C program to perform Addition of Two Matrices using functions.

c. Write a C program to implement Multiplication of Two Matrices

TASK 8

a. Write a C program that uses non-recursive function to search for a Key value in a given list of integers using linear search method.

b. Write a C program that uses non-recursive function to search for a Key value in a given sorted list of integers using binary search method.

TASK 9

a. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

b. Write a C program that sorts the given array of integers using selection sort in descending order.

c. Write a C program that sorts the given array of integers using insertion sort in ascending order.

TASK 10

a. Write a C program that uses functions to perform the following operations:

I. To insert a sub-string into a given main string from a given position.

II. To delete n Characters from a given position in a given string

b. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)

TASK 11

a. Write a C program to sort the ‘n’ strings in the alphabetical order using functions.

b. Write a C program to count the lines, words and characters in a given text.

TASK 12

a. Write a C program to implement function pointer to find sum and product of two numbers.

b. Write a program for reading elements using a pointer into an array and display the values using the array.

c. Write a program for display values reverse order from an array using a pointer.

TASK 13

a. Define a structure Date with members day, month, and year. Create another structure Employee with members: emp_id, emp_name, and a nested structure Date for joining_date. Write a program to store details of 5 employees in an array of structures and display employees who joined after the year 2020.

b. Write a C program that uses structures and functions to perform addition and product of two complex numbers? (use structures and functions)

TASK 14

a. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents

TASK 15

a. Write a C program to find sum of ‘n’ numbers using command line arguments.

b. Write a C program to implement following pre-processor directives:

i. define ii. undef iii. ifdef iv. ifndef.

c. Write a C program to create a user defined header file to find sum, product and greatestof two numbers.

TEXTBOOKS:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, CengageLearning, 3rd Edition.

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PrenticeHall of India.
2. R.G. Dromey, How to solve it by Computer, Pearson, 16th Impression.
3. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
4. Herbert Schildt, C: The Complete Reference, McGraw Hill, 4th Edition.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

Course Code: GR25A1019 **L/T/P/C: 0/0/2/1**

I Year I Semester

Course Outcomes: Students will be able to

1. Interpret the role and importance of various forms of communication skills.
2. Demonstrate the skills needed to participate in a conversation that builds knowledge collaboratively by listening carefully and respect others point of view.
3. Utilize various media of verbal and non-verbal communication with reference to various professional contexts.
4. Recognize the need to work in teams with appropriate ethical, social and professional responsibilities.
5. Speak and pronounce English intelligibly.

English Language and Communication Skills Lab (ELCS) shall have two parts:

a. **Computer Assisted Language Learning (CALL) Lab which focuses on listening skills**
b. **Interactive Communication Skills (ICS) Lab which focuses on speaking skills**

Exercise – I
CALL Lab:
Instruction: Speech Sounds-Listening Skill - Importance – Purpose - Types- Barriers-Active Listening
Practice: Listening to Distinguish Speech Sounds (Minimal Pairs) - *Testing Exercises*
ICS Lab:

❖ **Diagnostic Test – Activity titled '*Express Your View*'**

Instruction: Spoken and Written language - Formal and Informal English - Greetings - Introducing Oneself and Others
Practice: Any Ice-Breaking Activity

Exercise – II
CALL Lab:
Instruction: Listening vs. Hearing - Barriers to Listening
Practice: Listening for General Information - Multiple Choice Questions - *Listening Comprehension Exercises (It is essential to identify a suitable passage with exercises for practice.)*
ICS Lab:
Instruction: Features of Good Conversation – Strategies for Effective Communication
Practice: Role Play Activity - Situational Dialogues –Expressions used in Various Situations –Making Requests and Seeking Permissions – Taking Leave - Telephone Etiquette

Exercise – III
CALL Lab:

Instruction: Errors in Pronunciation – Tips for Neutralizing Mother Tongue Influence (MTI)

Practice: Differences between British and American Pronunciation –*Listening Comprehension Exercises*

ICS Lab:

Instruction: Describing Objects, Situations, Places, People and Events

Practice: Picture Description Activity – Looking at a Picture and Describing Objects, Situations, Places, People and Events *(A wide range of Materials / Handouts are to be made available in the lab.)*

Exercise – IV
CALL Lab:

Instruction: Techniques for *Effective* Listening

Practice: Listening for Specific Details - Listening - Gap Fill Exercises - *Listening Comprehension Exercises*

(It is essential to identify a suitable passage with exercises for practice.)

ICS Lab:

Instruction: How to Tell a Good Story - Story Star- Sequencing-Creativity

Practice: Activity on Telling and Retelling Stories - Collage

Exercise – V
CALL Lab:

Instruction: Identifying the literal and implied meaning

Practice: Listening for Evaluation - Write the Summary – Listening Comprehension Exercises

(It is essential to identify a suitable passage with exercises for practice.)

ICS Lab:

Instruction: Understanding Non-Verbal Communication

Practice: Silent Speech - Dumb Charades Activity

- **Post-Assessment Test on ‘Express Your View’**

Minimum Requirement of infrastructural facilities for ELCS Lab:

1. **Computer Assisted Language Learning (CALL) Lab:**

The Computer Assisted Language Learning Lab has to accommodate 40 students with 40 systems, with one Master Console, LAN facility and English language learning software for self- study by students.

System Requirement (Hardware component):

Computer network with LAN facility (minimum 40 systems with multimedia) with the following specifications:

i) Computers with Suitable Configuration

ii) High Fidelity Headphones

2. **Interactive Communication Skills (ICS) Lab:**

The Interactive Communication Skills Lab: A Spacious room with movable chairs and audio- visual aids with a Public Address System, a T. V. or LCD, a digital stereo – audio & video system and camcorder etc.

□ **Note: English Language Teachers are requested to prepare Materials / Handouts for each Activity for the Use of those Materials in CALL & ICS Labs.**

Suggested Software:

- Cambridge Advanced Learners' English Dictionary with CD.
- Grammar Made Easy by Darling Kindersley.
- Punctuation Made Easy by Darling Kindersley.
- Oxford Advanced Learner's Compass, 10th Edition.
- English in Mind (Series 1-4), Herbert Puchta and Jeff Stranks with Meredith Levy, Cambridge.
- English Pronunciation in Use (Elementary, Intermediate, Advanced) Cambridge University Press.
- English Vocabulary in Use (Elementary, Intermediate, Advanced) Cambridge University Press.
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS).

REFERENCE BOOKS:

1. Shobha, KN & Rayen, J. Lourdes. (2019). *Communicative English – A workbook.* Cambridge University Press
2. Board of Editors. (2016). *ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities.* Orient BlackSwan Pvt. Ltd.
3. Mishra, Veerendra et al. (2020). *English Language Skills: A Practical Approach.* Cambridge University Press
4. (2022). *English Language Communication Skills – Lab Manual cum Workbook.* Cengage Learning India Pvt. Ltd.
5. Ur, Penny and Wright, Andrew. 2022. *Five Minute Activities – A Resource Book for Language Teachers.* Cambridge University Press.

I YEAR
II SEMESTER

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS

Course Code: GR25A1002 **L/T/P/C: 3/1/0/4**

I Year II Semester

Pre-requisites: Mathematical Knowledge at pre-university level

Course Outcomes: After learning the contents of this course, the student must be able to

1. Identify whether the given differential equation of first order is exact or not
2. Solve higher differential equation and apply the concept of differential equation to real world problems.
3. Use the Laplace Transforms techniques for solving Ordinary Differential Equations.
4. Evaluate the line integrals and use them to calculate work done
5. Evaluate surface and volume integrals and apply fundamental theorems of vector calculus to relate line integrals and surface integrals

UNIT-I : First Order Ordinary Differential Equations

Exact differential equations – Equations reducible to exact differential equations – linear and Bernoulli's equations –Applications : Newton's law of cooling – Law of natural growth and decay.- modelling of R-L circuit and R-C Circuit

UNIT-II: Ordinary Differential Equations of Higher Order

Higher order linear differential equations with constant coefficients: Non-Homogeneous terms of the type e^{ax}, sin , cos ax, polynomials in x, $e^{ax}V(x)$ and $x\ V(x)$ – Method of variation of parameters.

UNIT-III : Laplace Transforms

Laplace Transforms: Laplace Transform of standard functions – First shifting theorem – Laplace transforms of functions multiplied by 't' and divided by 't' – Laplace transforms of derivatives and integrals of function – Inverse Laplace transform by different methods, Applications **:** solving Initial value problems by Laplace Transform method.

UNIT-IV : Vector Differentiation and Line Integration

Vector differentiation **:** Scalar and vector point functions, Concepts of gradient, Directional derivatives , divergence and curl of functions in cartesian framework- solenoidal field, irrotational field , scalar potential

Vector line integration **:** Evaluation of the line integral, concept of work done by a force field, Conservative fields

UNIT-V :Surface Integration And Vector Integral Theorems

Surface integration: Evaluation of surface and volume integrals, flux across a surface

Vector integral theorems: Green's, Gauss and Stokes theorems (without proof) and their applications

TEXTBOOKS :

1. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. R.K. Jain and S.R.K. Iyengar, Advanced Engineering Mathematics, Narosa Publications, 5th Edition, 2016.

REFERENCE BOOKS :

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9thEdition, Pearson, Reprint, 2002.
3. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
4. H. K. Dass and Er. Rajnish Verma, Higher Engineering Mathematics, S Chand and Company Limited, New Delhi.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGINEERING CHEMISTRY

Course Code: GR25A1004 **L/T/P/C: 3/0/0/3**

I Year II Semester

Course Outcomes: After completion of the course, the student should be able to

1. Understand the specifications, water quality and treatment methods for domestic & Industrial needs.
2. Apply electrochemical concepts and analyze corrosion processes with suitable control measures.
3. Distinguish various energy sources to prioritize eco-friendly fuels for environmental sustainable development.
4. Analyse the efficacy of polymers in diverse applications
5. Interpret the role of engineering materials and emphasize the scope of spectroscopic techniques in various sectors.

UNIT-I: Water and its treatment:

Introduction- Hardness, types, degree of hardness and units. Estimation of hardness of water by complexometric method - Numerical problems. Potable water and its specifications (WHO) - Steps involved in the treatment of potable water - Disinfection of potable water by chlorination and breakpoint chlorination. Defluoridation - Nalgonda technique.

Boiler troubles: Scales, Sludges and Caustic embrittlement. Internal treatment of boiler feed water - Calgon conditioning, Phosphate conditioning, Colloidal conditioning.

External treatment methods - Softening of water by ion- exchange processes. Desalination of brackish water – Reverse osmosis.

UNIT-II: Electrochemistry and Corrosion:

Electrode potential, standard electrode potential, Nernst equation (no derivation), electrochemical cell - Galvanic cell, cell representation, EMF of cell - Numerical problems. Types of electrodes, reference electrodes - Primary reference electrode - Standard Hydrogen Electrode (SHE), Secondary reference electrode - Calomel electrode. Construction, working and determination of pH of unknown solution using SHE and Calomel electrode.

Corrosion: Definition, causes and effects of corrosion – Theories of corrosion, chemical and electrochemical theories of corrosion, Types of corrosion: galvanic, water-line and pitting corrosion. Factors affecting rate of corrosion - Nature of the metal, Nature of the corroding environment. Corrosion control methods - Cathodic protection Methods - Sacrificial anode and impressed current methods.

UNIT–III: Energy sources

Batteries: Definition – Classification of batteries - Primary, secondary and reserve batteries with examples. Construction, working and applications of Zn-air and Lithium ion battery. Fuel Cells – Differences between a battery and a fuel cell, Construction and applications of Hydrogen –Oxygen Fuel Cell.

Fuels: Definition and characteristics of a good fuel, Calorific value – Units - HCV, LCV- Dulongs formula - Numerical problems.

Fossil fuels: Classification, Petroleum - Refining of Crude oil, Cracking - Types of cracking

- Moving bed catalytic cracking. LPG and CNG composition and uses. Synthetic Fuels: Fischer Tropschs Process, Introduction and applications of Hythane and Green Hydrogen.

UNIT - IV: Polymers

Definition - Classification of polymers: Based on origin and tacticity with examples – Types of polymerization - Addition (free radical addition mechanism) and condensation polymerization.

Plastics and Fibers: Definition and applications (PVC, Nylon-6,6). Differences between themoplastics and thermo setting plastics, Fiber reinforced plastics (FRP).

Conducting polymers: Definition and Classification with examples - Mechanism of conduction in transpoly-acetylene and applications of conducting polymers.

Biodegradable polymers: Polylactic acid and its applications.

UNIT-V- Engineering Materials

Smart materials: Classification with examples - Shape Memory Alloys – Nitinol, Piezoelectric materials – quartz and their engineering applications.

Biosensor - Definition, Amperometric Glucose monitor sensor.

Cement: Portland cement, its composition, setting and hardening.

Interpretative spectroscopic applications of UV-Visible spectroscopy for Analysis of pollutants in dye industry, IR spectroscopy in night vision-security, Pollution Under Control- CO sensor (Passive Infrared detection).

TEXTBOOKS:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010.
2. Engineering Chemistry by Rama Devi, Dr. P. Aparna and Rath, Cengage learning, 2025.

REFERENCE BOOKS:

1. Engineering Chemistry: by Thirumala Chary Laxminarayana & Shashikala, Pearson Publications (2020).
2. Engineering Chemistry by Shashi Chawla, Dhanpatrai and Company (P) Ltd. Delhi 2011.
3. Engineering Chemistry by Shikha Agarwal, Cambridge University Press, Delhi 2015.
4. Engineering Analysis of Smart Material Systems by Donald J. Leo, Wiley, 2007.
5. Challenges and Opportunities in Green Hydrogen by Editors: Paramvir Singh, Avinash Kumar Agarwal, Anupma Thakur, R.K Sinha.

E-books:

https://archive.org/details/EngineeringChemistryByShashiChawla/page/n11/mode/2u

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

FUNDAMENTALS OF ELECTRICAL ENGINEERING

Course Code: GR25A1007 **L/T/P/C: 3/0/0/3**

I Year II Semester

COURSE OUTCOMES

1. Summarize the basic fundamental laws of electric circuits.
2. Analyze electric circuits with suitable theorems.
3. Distinguish the single phase and three phase systems.
4. Interpret the working principle of Electrical machines.
5. Outline the protection principles using Switchgear components.

UNIT I - NETWORK ELEMENTS & LAWS

Charge, Current, Voltage, Power, Active elements, Independent and dependent sources. Passive elements - R, L and C, Energy stored in inductance and capacitance, Kirchhoff's laws, Source transformations, Star-delta transformations, Node voltage method, mesh current method.

UNIT II - NETWORK THEOREMS

Superposition theorem, Thevenin's theorem, Norton's theorems, Maximum power transfer theorem and Reciprocity theorem (DC Circuits).

UNIT III - AC CIRCUITS

Representation of sinusoidal waveforms, RMS and average values of periodic sinusoidal and non- sinusoidal waveforms, Phasor representation, Types of power, active power, Reactive power and Apparent power, Power factor. Impedance and Admittance, Analysis of series, parallel and series-parallel circuits, Introduction to three-phase circuits, types of connection. voltage and current relations in star and delta connections. Resonance: Series circuits, Bandwidth and Q-factor.

UNIT IV- BASICS OF ELECTRICAL MACHINES

Transformer: Mutual Induction, construction and working principle, Types of transformers, Ideal transformer, EMF Equation-simple Problems.

Construction and working principles of DC generator, DC motor, Synchronous generator, and Induction Motor – applications.

UNIT V - ELECTRICAL INSTALLATIONS

Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, MCCB, Earthing – Plate and Pipe Earthing. Types of Batteries – Primary and Secondary, UPS (Uninterrupted power supply)-components, calculation of ratings for UPS-Components to a specific load, power factor improvement methods.

TEXTBOOKS

1. "Basic Electrical Engineering", D.P. Kothari and I.J. Nagrath, Third edition 2010, Tata McGraw Hill.
2. "Electrical Engineering Fundamentals", Vincent Deltoro, Second Edition, Prentice Hall India, 1989.

REFERENCE BOOKS:

1. "A Textbook of Electrical Technology",- BL Theraja volume-I, S.Chand Publications.
2. "Electrical Machinery", P. S. Bimbhra, Khanna Publishers, 2011.
3. "Electrical and Electronics Technology", E. Hughes, 10th Edition, Pearson, 2010.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA STRUCTURES AND ALGORITHMS

Course Code: GR25A1028 **L/T/P/C: 2/ 0/ 0/ 2**

I Year II Semester

Course Prerequisites: C Language

Course Outcomes:

1. Analyze the computational complexity of algorithms using asymptotic notations to evaluate their efficiency.
2. Implement fundamental linear and non-linear data structures and perform essential operations on them.
3. Apply graph traversal algorithms and tree operations to solve complex computational problems.
4. Select and implement appropriate searching and sorting algorithms for a given task.
5. Design efficient software solutions by choosing optimal data structures and algorithms based on problem constraints.

UNIT I

Basic Terminologies & Introduction to Algorithm and Data Organization: Algorithm specification, Recursion, Performance analysis, Asymptotic Notation - The Big-O,Omega and Theta notation, Programming style, Refinement of coding-Time-Space Trade Off, Testing, Data Abstraction

UNIT II

Linear Data Structure: Array**,** Stack, Queue, Linked list and its types, Various Representations, Operations & Applications of Linear Data Structures.

UNIT III

Non-linear Data Structure: Trees: Binary Tree – Terminology and basic operations (no implementation), Binary Search Tree – Insert, delete, search, traversal and implementation, B Tree, B+ Tree, AVL Tree, Splay Tree (B, B+, AVL trees onlydefinitions no implementation).

UNIT IV

Non-linear Data Structure: Graphs: Basic Terminologies, Directed, Undirected and Representations, Graph search and Traversal algorithms Breadth First Search, Depth First Search and complexity analysis, Applications of Non-Linear Data Structures.

UNIT V

Searching and Sorting on Various Data Structures: Sequential Search, Binary Search, Insertion Sort, Selection Sort, Shell Sort, Heap Sort, Divide and Conquer Sort: Merge Sort, Quick Sort, Comparison Trees (Decision tree), Introduction to Hashing.

File: Organization Sequential, Direct, Indexed Sequential, Hashed, and various types of accessing schemes (no implementation).

TEXTBOOKS:

1. Fundamentals of Data Structures, E. Horowitz and S. Sahni, 1977.
2. Data Structures and Algorithms, Alfred V. Aho, John E. Hopperoft, Jeffrey,D. UIlman.

REFERENCE BOOKS:

1. The Art of Computer Programming: Volume 1: Fundamental Algorithms, DonaldE. Knuth
2. Introduction to Algorithms, Thomas, H. Cormen, Charles E. Leiserson, Ronald
3. L. Rivest, Clifford Stein.Open Data Structures: An Introduction (Open Paths to Enriched Learning), 31st Edition, Pat Morin.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

PYTHON PROGRAMMING

Course Code: GR25A1026 **L/T/P/C: 1/0/0/1**

I Year II Semester

Course Outcomes:

1. Demonstrate the fundamental concepts and flow control in Python
2. Implement different sequence types and file handling operations.
3. Design python programs using functions and exception handling mechanisms.
4. Develop programs with object oriented programming features and modules.
5. Design GUI based applications using Tkinter.

UNIT I

Introduction: features of Python-Interactive execution, comments, types, variables, operators, expressions, Statements-assignment, input, print.

Control flow: if, if-else, if-elif-else Statements, Nested Decision Structures, Loops- while loop, for loop, Nested Loops, break, continue, pass statement.

UNIT II

Sequences: Strings, Lists and Tuples-basic operations and functions, iterating over sequences, Sets and Dictionaries- operations and functions, Python program examples.

Files-operations-opening, reading, writing, closing, file positions.

UNIT III

Exceptions: raising and handling exceptions, try/except statements, finally clause, standard exceptions, custom exceptions.

Functions: definition, call, scope and lifetime of variables, keyword arguments, default parameter
values, variable length arguments, recursive functions, Lambda function.

UNIT IV

Modules: Modules, Standard Modules, Importing Modules, Namespaces and Packages.

Object Oriented Programming: Classes, constructors, objects, class variables, class methods, static methods, operator overloading. Inheritance-is-a relationship, composition, polymorphism, overriding, multiple inheritance, abstract classes, multithreaded programming, Python program examples.

UNIT V

GUI Programming: Introduction, Tkinter, Widgets (Buttons, Canvas, Frame, Label, Menu, Entry, Text, Scrollbar, Combobox, Listbox), event driven programming-events, callbacks, binding, layout management- geometry managers: pack and grid, creating GUI based applications in Python.

Teaching methodologies:

- Power Point Presentations
- Tutorial Sheets
- Assignments

TEXTBOOKS:

1. Exploring Python, Timothy A. Budd, McGraw Hill Publications.
2. Introduction to Programming using Python, Y.Daniel Liang, Pearson.
3. Python Programming, Sheetal Taneja and Naveen Kumar, Pearson.

REFERENCE BOOKS:

1. Introduction to Computer Science using Python, Charles Dierbach, Wiley India Edition.
2. Internet of Things - A hands on approach, Arshdeep Bahga and Vijay Madisetti,Universities Press, 2015.
3. Fundamentals of Python, K. A. Lambert, B.L. Juneja, Cengage Learning.
4. Think Python, how to think like a computer scientist, Allen B. Downey,SPD, O'Reilly.
5. Core Python Programming, Wesley J.Chun, second edition, pearson.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

INNOVATION AND DESIGN THINKING

Course Code: GR25A1027 **L/T/P/C: 1/0/0/1**

I Year II Semester

Course Outcomes (COs): On successful completion of the course, the student will be able to:

1. Explain the concepts and importance of innovation and design thinking.
2. Apply industry analysis tools and ideation techniques to identify problems and opportunities.
3. Develop prototypes and assess market potential for innovative ideas.
4. Demonstrate understanding of sustainable design models and their applications.
5. Describe the basics of IPR and apply them in protecting and managing innovations.

UNIT I - Fundamentals of Design Thinking and Innovation

Design Thinking: Meaning and definition of Design Thinking, Nature, features, and importance of Design Thinking.

Principles of Design Thinking (Empathy, Define, Ideate, Prototype, Test) Design Thinking mind set and skills required. Difference between Design Thinking and traditional problem-solving. Applications of Design Thinking in business.

Innovation: Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation.

Core Teaching Tool: Simulation, Game, Industry Case Studies (Personalized for students – 16 industries to choose from).

UNIT II - Innovation through Opportunity Mapping and Design Thinking

Competition and Industry trends mapping and assessing initial opportunity, Porter’s Five Force Model. Identification of gap, problem, analysing the problem from an industry perspective. Idea generation.

Ideation techniques: Brainstorming, Brain writing, Round robin, and SCAMPE. Mapping of solution to problem: Problem–Solution Fit, Steps in Mapping.

Tools and Techniques of mapping : (Value Proposition Canvas, Problem–Solution Matrix, User Journey Mapping, Prototyping and testing for validation).

Core Teaching Tool: Several types of activities including Class, game, Gen AI, Journey Mapping Exercise (Pick a common activity (e.g., ordering food online, booking tickets, paying college fees) Students map the customer journey step by step, highlighting touch points and problems faced at each stage.

UNIT III - Opportunity assessment and Prototype development

Identify and map global competitors, review industry trends, and understand market sizing: TAM (Total Addressable Market), SAM (Serviceable Available Market) and SOM (Serviceable Obtainable Market). Assessing scope and potential scale for the opportunity.

Understanding prototyping and Minimum Viable Product (MVP).

Developing a prototype: Testing, and validation.

Core Teaching Tool: Venture Activity for prototype, no-code Innovation tools, Class activity.

UNIT IV- Sustainable Design Approaches / Models:

Introduction to Sustainable Design – Meaning, importance, and role in today's context. **Principles of Sustainable Design** (Reduce, Reuse, Recycle, Circular Economy, Cradle-to-Cradle approach).

Models of Sustainable Design: Product Life Cycle Design (from raw material to disposal), Eco-Design Model, Systems Thinking Approach.

Strategies for Sustainable Design: Green materials, energy efficiency, waste reduction, ethical sourcing. Applications – Sustainable product and service design.

Core Teaching Tool: Case Studies – Examples from industries adopting sustainable design

UNIT V

IPR Management: Meaning and importance of Intellectual Property (IP)

Types of Intellectual Property: Patents, Trademarks, Copyrights, Industrial Designs, Trade Secrets, Geographical Indications. Role of IPR in innovation and technology development.

Patents and Patent System: Scope and criteria for patentability (novelty, utility, non-obviousness), Procedure for grants of patents. Indian Scenario of Patenting.

IPR Management in Engineering: Protecting innovations: Licensing, Technology transfer, Commercialization, infringement issues. Emerging issues: IPR in Artificial Intelligence, Biotechnology, Software, and Digital Platforms.

Core Teaching Tool: Expert talks; Cases; Class activity and discussions; Venture Activities.

TEXTBOOKS :

1. A Textbook on Design Thinking: Principles, Processes & Applications – Srinivasan R., Mohammed Ismail, and Arulmozhi Srinivasan, S. Chand Publishing, 2025.
2. Design Thinking: A Comprehensive Textbook – Shalini Rahul Tiwari and Rohit Rajendra Swarup, Wiley India, 2024.
3. Design Thinking for Engineering: A Practical Guide – Edited by Iñigo Cuiñas and Manuel José Fernández Iglesias, Institution of Engineering and Technology (IET), 2023.
4. Management of Innovation and Product Development: Integrating Business and Technological Perspectives – Marco Cantamessa and Francesca Montagna, Springer London, 2023.
5. Managing Innovation: Integrating Technological, Market and Organizational Change (8th Edition) – Joe Tidd and John Bessant, Wiley, Latest Edition.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGINEERING WORKSHOP

Course Code: GR25A1024 **L/T/P/C: 1/0/3/2.5**

I Year II Semester

Course Outcomes

1. Identify workshop tools and their operational capabilities
2. Practice on manufacturing components using workshop trades including Carpentry, Fitting, Tin Smithy, Welding, Foundry and Black Smithy
3. Apply basic electrical engineering knowledge for House Wiring Practice
4. Develop various trades applicable to industries
5. Create hands on experience for common trades with taking safety precautions

TRADES FOR EXERCISES: At least two tasks from each trade

1. Carpentry: Demonstration and practice of carpentry tools

Task 1: Preparation of T- Lap Joint
Task 2: Preparation of Dove Tail Joint.

2. Fitting - Demonstration and practice of fitting tools
Task 3: Preparation of Straight Fit
Task 4: Preparation of V-Fit

3. Tin-Smithy - Demonstration and practice of Tin Smithy tools
Task 5: Preparation of Rectangular Tray
Task 6: Preparation of Open Scoop

4. Welding : Demonstration and practice on Arc Welding tools
Task 7: Preparation of Lap joint,
Task 8: Preparation of Butt Joint

5. House-wiring: Demonstration and practice on House Wiring tools
Task 9: Exercise on One way switch controlled two bulbs in series one bulb in Parallel.
Task 10: Exercise on Staircase connection.

6. Foundry : Demonstration and practice on Foundry tools
Task 11: Preparation of Mould using Single Piece Pattern.
Task 12: Preparation of Mould using Split Piece Pattern.

7. Black Smithy: Demonstration and practice on Black Smithy tools
Task 13: Preparation of U-Hook
Task 14: Preparation of S-Hook

TRADES FOR DEMONSTRATION: Plumbing, Machine Shop, Power tools in construction and Wood Working

Preparation of a prototype model of any trade under G-LOB activity

TEXTBOOKS:

1. Basic Workshop Technology: Manufacturing Process, Felix W.; Independently Published,2019.
2. Workshop Processes, Practices and Materials; Bruce J. Black, Routledge publishers, 5thEdn. 2015.
3. A Course in Workshop Technology Vol I. & II, B.S. Raghuwanshi, Dhanpath Rai & Co., 2015 & 2017.

REFERENCE BOOKS:

1. Elements of Workshop Technology, Vol. I by S. K. Hajra Choudhury & Others, Media Promoters and Publishers, Mumbai. 2007, 14th edition
2. Elements of Workshop Technology, Vol. II by S. K. Hajra Choudhury & Others, Media Promoters and Publishers, Mumbai. 2007, 12th edition
3. Workshop Practice by H. S. Bawa, Tata-McGraw Hill, 2004.
4. Technology of machine tools, Steve F. Krar, Arthur R. Gill and Peter Smid, McGraw Hill Education (India) Pt. Ltd., 2013.
5. Engineering Practices Laboratory Manual, Ramesh Babu.V., VRB Publishers Private Limited, Chennai, Revised edition, 2013 – 2014.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGINEERING CHEMISTRY LAB

Course Code: GR25A1018 **L/T/P/C: 0/0/2/1**

I Year II Semester

Course Outcomes:

1. Identify key water quality parameters such as hardness, chloride by volumetric analysis.
2. Apply analytical techniques such as conductometry to estimate acids, and colorimetry to validate theoretical principles like Beer–Lambert's law.
3. Determine the concentrations of acids, base, and ferrous ions by potentiometric titration.
4. Synthesize polymers like Bakelite and Nylon-6,6 to gain practical experience.
5. Estimate the physicochemical properties of materials such as viscosity, acid value, and corrosion rate**.**

List of Experiments:

1. Estimation of Hardness of water by EDTA Complexometric method.
2. Determination of chloride content of water by Argentometric method.
3. Estimation of the concentration of a strong acid by Conductometry.
4. Estimation of the concentration of strong and weak acids in an acid mixture by Conductometry.
5. Estimation of the concentration of Fe^{+2} ion by Potentiometry using $K_2Cr_2O_7$.
6. Estimation of the concentration of a strong acid with a strong base by Potentiometry using quinhydrone.
7. Colorimetric analysis of Potassium Permanganate: Verification of Beer–Lambert's Law.
8. Preparations:
    a. Preparation of Bakelite.
    b. Preparation Nylon – 6, 6.
9. Determination of corrosion rate of mild steel in the presence and absence of inhibitor.
10. Estimation of the acid value of the given lubricant oil.
11. Estimation of viscosity of lubricant oil using Ostwald's Viscometer.
12. **Virtual Labs:**
    a. Construction of Fuel cell and it's working.
    b. Smart materials for Biomedical applications
    c. Batteries for electrical vehicles.
    d. Functioning of solar cell and its applications.

Textbooks:

1. Vogel's text book of Practical organic chemistry, 8th Edition.

Reference Books:

1. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, S Chand Publications, New Delhi (2022)
2. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.
3. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007)

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

FUNDAMENTALS OF ELECTRICAL ENGINEERING LAB

Course Code: GR25A1022 **L/T/P/C: 0/0/2/1**

I Year II Semester

Course Outcomes:

1. Demonstrate the common electrical components and their ratings.
2. Summarize the basic fundamental laws of electric circuits.
3. Distinguish the measurement and relation between the basic electrical parameters
4. Examine the response of different types of electrical circuits with three phase excitation.
5. Outline the basic characteristics of Electrical machines.

LIST OF EXPERIMENTS

Any ten experiments should be conducted.

1. Verification of Ohms Law, KVL and KCL.
2. Verification of Thevenin’s & Norton’s Theorems.
3. Verification of Superposition and Reciprocity Theorems.
4. Resonance in series RLC circuit.
5. Load Test on Single Phase Transformer (Calculate Efficiency and Regulation)
6. Verification of Voltage and Current relations in Three Phase Circuits (Star-Delta)
7. Measurement of Active and Reactive Power in a balanced Three-phase circuit.
8. Torque – speed characteristics of a Separately Excited DC Shunt Motor.
9. Torque-Slip Characteristics of a Three-phase Induction Motor.
10. No-Load Characteristics of a Three-phase Alternator.
11. Verification of Maximum Power Transfer Theorem.
12. Power factor improvement by using capacitor bank in parallel with inductive load.

TEXTBOOKS

1. “Basic Electrical Engineering”, D.P. Kothari and I.J. Nagrath, Third edition 2010, Tata McGraw Hill.

2. “Electrical Engineering Fundamentals”, Vincent Deltoro, Second Edition, Prentice Hall India, 1989.

REFERENCE BOOKS:

1. “A Textbook of Electrical Technology”,- BL Theraja volume-I, S.Chand Publications.
2. “Electrical Machinery”, P. S. Bimbhra, Khanna Publishers, 2011.
3. “Electrical and Electronics Technology”, E. Hughes, 10th Edition, Pearson, 2010.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA STRUCTURES AND ALGORITHMS LAB

Course Code: GR25A1029 **L/T/P/C: 0/0/2/1**

I Year II Semester

Prerequisites: Basic Programming fundamentals

Course Outcomes:

1. Implement operations on various linear and non-linear data structures.
2. To identify the appropriate data structure for solving a given problem.
3. Acquire practical knowledge on applications of various data structures.
4. Implement various searching and sorting techniques.
5. To effectively troubleshoot, debug and run programs in C.

LIST OF EXPERIMENTS:

TASK-1: Fundamental Recursion and Linear Data Structures

a) Towers of Hanoi

- **Experiment:** Implement recursive solution for moving disks between rods.
- **Real Example:** Algorithmic Puzzle
  - Disks represent problems of decreasing size.
  - Move disks from source to target rod using auxiliary rod.
  - Solve smaller sub problems recursively.
- **Use Case:** Constraint satisfaction problems, recursive algorithm design.

b) Stack using Arrays

- **Experiment:** Implement push, pop, and display operations using arrays.
- **Real Example:** Undo/Redo Feature
  - Each edit operation is pushed onto stack.
  - Pop operation reverts to previous state.
  - Maintain history of actions for reversal.
- **Use Case:** Text editors, graphic software, browser history.

c) Queue using Arrays

- **Experiment:** Implement enqueue, dequeue, and display operations.
- **Real Example:** Print Spooler
  - Print jobs are added to queue.
  - Jobs processed in FIFO order.
  - Maintain order of job submissions.
- **Use Case:** Print systems, task scheduling, message queues.

TASK-2: Expression Evaluation and Circular Buffers

a) Evaluate a Postfix Expression

- **Experiment:** Implement stack-based postfix expression evaluation.
- **Real Example:** Calculator Arithmetic
  - Operands are pushed onto stack.
  - Operators applied to top stack elements.
  - Result pushed back for further operations.
- **Use Case:** Compilers, calculators, expression parsing.

b) Circular Queue using Arrays

- **Experiment:** Implement circular enqueue and dequeue operations.
- **Real Example:** Data Streaming Buffer
  - Continuous data overwrites oldest entries.
  - Efficient memory usage for real-time data.
  - Handles continuous input streams.
- **Use Case:** Audio/video streaming, network packets, sensor data.

TASK-3: Polynomial Arithmetic

a) Reading, Writing, and Addition of Polynomials

- **Experiment:** Implement polynomial storage and addition operations.
- **Real Example:** Symbolic Mathematics
  - Terms stored with coefficients and exponents.
  - Like terms combined during addition.
  - Results displayed in standard form.
- **Use Case:** Computer algebra systems, engineering calculations.

TASK-4: Doubly Linked List Operations

a) Create, Insert, Delete, Search, and Traversal

- **Experiment:** Implement bidirectional linked list operations.
- **Real Example:** Media Playlist Navigation
  - Songs can be traversed forward and backward.
  - Dynamic addition and removal of tracks.
  - Current track maintained with neighbours.
- **Use Case:** Music players, browser history, navigation systems.

TASK-5: Binary Search Tree Operations

a) Insert, Delete, Search in a BST

- **Experiment:** Implement BST operations maintaining sort property.
- **Real Example:** Dictionary System
  - Words stored in sorted order.
  - Fast lookup using binary search.
  - Dynamic updates to vocabulary.
- **Use Case:** Spell checkers, databases, auto complete systems.

TASK-6: Tree Traversal Techniques

a) BFS and DFS Traversal

- **Experiment:** Implement level-order and depth-first traversals.
- **Real Example:** File System Navigation
  - BFS lists files level by level.
  - DFS explores directory branches deeply.
  - Complete system structure exploration.
- **Use Case:** File managers, system utilities, data analysis.

TASK-7: Graph Traversal Algorithms

a) Breadth First Search (BFS) on Graphs

- **Experiment:** Implement BFS using queue for graph traversal.
- **Real Example:** Network Routing
  - Finds shortest path in unweighted graphs.
  - Explores all neighbours at current depth.
  - Useful for network broadcasting.
- **Use Case:** GPS navigation, social networks, web crawling.

b) Depth First Search (DFS) on Graphs

- **Experiment:** Implement DFS using stack or recursion.
- **Real Example:** Dependency Resolution
  - Detects cycles in dependencies.
  - Topological sorting for task ordering.
  - Path finding in maze-like structures.
- **Use Case:** Compilers, project scheduling, game AI.

TASK-8: Search Algorithms

a) Sequential Search

- **Experiment:** Implement linear search through collections.
- **Real Example:** Simple List Lookup
  - Checks each element sequentially.
  - Works on unsorted data.
  - Simple but inefficient for large data.
- **Use Case:** Small lists, unsorted data, simple applications.

b) Binary Search

- **Experiment:** Implement divide-and-conquer search algorithm.
- **Real Example**: Phone Directory Search
  - Requires sorted data.
  - Halves search space each iteration.
  - Extremely efficient for large datasets.
- **Use Case:** Databases, libraries, sorted collections.

TASK-9: Basic Sorting Algorithms

a) Insertion Sort

- **Experiment:** Implement sorting by inserting elements in position.

- **Real Example:** Card Player Sorting
    - Builds sorted portion incrementally.
    - Efficient for small or nearly sorted data.
    - Simple implementation.
- **Use Case:** Small datasets, educational purposes.

b) Selection Sort

- **Experiment:** Implement sorting by repeatedly finding minimum element.
- **Real Example:** Contestant Ranking
    - Selects best element each round.
    - Simple but inefficient for large data.
    - Minimal memory usage.
- **Use Case:** Educational demonstrations, small collections.

TASK-10: Advanced Sorting Algorithms

a) Shell Sort

- **Experiment:** Implement gap-based insertion sort variant.
- **Real Example:** Library Catalog Sorting
    - Sorts subarrays with decreasing gaps.
    - More efficient than basic insertion sort.
    - Good for medium-sized data.
- **Use Case:** Medium collections, practical sorting applications.

b) Heap Sort

- **Experiment:** Implement sorting using heap data structure.
- **Real Example:** Priority Task Scheduling
    - Builds max-heap for sorting.
    - In-place sorting algorithm.
    - Guaranteed O(n log n) performance.
- **Use Case:** Systems programming, priority queues.

TASK-11: Divide and Conquer Sorting

a) Merge Sort

- **Experiment:** Implement recursive divide-and-conquer sorting.
- **Real Example**: Large File Sorting
    - Divides array into halves recursively.
    - Merges sorted halves efficiently.
    - Stable sorting algorithm.
- **Use Case:** External sorting, database systems.

b) Quick Sort

- **Experiment:** Implement partitioning-based sorting algorithm.
- **Real Example:** In-Memory Data Sorting
    - Selects pivot and partitions array.
    - Recursively sorts partitions.
    - Average-case efficient.
- **Use Case:** Programming language libraries, general-purpose sorting.

TASK-12: File Handling and Text Processing

a) Line Editors with Line Count, Word Count

- **Experiment:** Implement text processing with statistics.
- **Real Example:** Document Analysis Tool
    - Reads text file line by line.
    - Counts lines, words, and characters.
    - Displays real-time statistics.
- **Use Case:** Text editors, document processors, coding environments.

b) Binary Search Tree File Operations

i. Construct a BST from a File

- **Experiment:** Implement BST creation from file data.
- **Real Example:** Data Restoration
    - Reads sorted data from file.
    - Rebuilds balanced BST structure.
    - Recovers previous data state.
- **Use Case:** Data persistence, configuration management.

ii. Display BST Contents to a File

- **Experiment:** Implement BST traversal to file output.
- **Real Example:** Data Backup System
    - Performs in-order traversal.
    - Writes sorted data to file.
    - Preserves data structure integrity.
- **Use Case:** Data archiving, transfer between systems**.**

TEXTBOOKS:

1. Fundamentals of Data Structures, E. Horowitz and S. Sahni, 1977.
2. Data Structures and Algorithms, Alfred V. Aho, John E. Hopperoft, Jeffrey D. UIlman.

REFERENCE BOOKS:

1. The Art of Computer Programming: Volume 1: Fundamental Algorithms, Donald,E. Knuth.
2. Introduction to Algorithms, Thomas, H. Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein.
3. Open Data Structures: An Introduction (Open Paths to Enriched Learning), 31st Edition, Pat Morin.

II YEAR
I SEMESTER

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DISCRETE MATHEMATICS

Course Code: GR25A2076 **L/T/P/C: 3/0/0/3**

II Year I Semester

Course Outcomes:

1. Use propositional and predicate logic in knowledge representation and truth verification.
2. Demonstrate the application of discrete structures in different fields of computer science.
3. Apply basic and advanced principles of counting to the real-world problems.
4. Formulate the problem and solve using recurrence relations and generating functions.
5. Devise the given problem as a graph network and solve with techniques of graph theory.

UNIT I

Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth tables, Tautology, Equivalence implication, Normal forms.
Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof ofcontradiction.

UNIT II

Set Theory: Properties of binary relations, Compatibility, Equivalence and Partial ordering relations, Hasse diagram, Lattice and its properties.
Functions: Inverse function, Composite of functions, Recursive functions, Pigeon hole principle and itsapplication.
Algebraic Structures: Algebraic systems examples and general properties, Semi groups and monads,groups and sub groups' Homomorphism, Isomorphism.

UNIT III

Elementary Combinatorics: Basics of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial coefficients, Binomial and Multinomial theorems, the principle of Inclusion – Exclusion.

UNIT IV

Recurrence Relation: Generating functions, Function of sequences calculating coefficient of generating function, Recurrence relations, solving recurrence relation by substitution, Generating functions and Characteristics roots, solution of Inhomogeneous recurrence relation.

UNIT V

Graph Theory: Representation of graph, Graph theory and applications, Planar graphs, basic concepts ofIsomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic numbers, Depth First Search, Breadth First Search, Spanning trees.

Teaching Methodologies:

- Power Point Presentations
- Tutorial Sheets
- Assignments

TEXTBOOKS:

1. Discrete and Combinational Mathematics- An Applied Introduction-5th Edition–Ralph.P.Grimadi
   .Pearson Education.
2. Discrete Mathematical Structures with applications to computer science Trembly J.P. &Manohar.P,TMH.
3. Discrete Mathematics for Computer Scientists and Mathematicians 2nd Edition by Joe L. Mott, Abhraham Kandel and Theodore P. Baker.

REFERENCE BOOKS:

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.
2. Discrete Mathematics with Applications, Thomas Koshy, Elsevier.
3. Discrete Mathematical Structures, Bernand Kolman, Roberty C. Busby, Sharn Cutter Ross,Pearson.
4. Discrete mathematical structures, Dr. D S Chandrashekar, PRISM Publishers.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code: GR25A2083 **L/T/P/C: 3/0/0/3**

II Year I Semester

Prerequisite:
Students should have prior knowledge of basic computers and programming fundamentals.

Course Outcomes:

1. Apply Boolean algebra and logic gates in the development of digital circuits.
2. Analyze combinational and sequential logic circuits.
3. Utilize register transfer language to represent operations within the instruction cycle.
4. Perform computer arithmetic operations and apply micro-programmed control concepts.
5. Analyze memory organization and input/output interactions with the CPU.

UNIT I

Boolean Algebra and Logic Gates: Binary codes, Binary Storage and Registers, Binary logic, Digital logic gates.

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

UNIT II

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure. Registers, shift Registers, Ripple counters, synchronous counters, other counters.

UNIT III

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro-operations, shift micro-operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT IV

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT V

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

TEXTBOOKS:

1. M. Morris Mano, Michael D. Ciletti, Digital Design, Sixth Edition, Pearson, 2021.
2. William Stallings, Computer Organization and Architecture, 11th Edition, Pearson/PHI, 2023.

REFERENCE BOOKS:

1. ZVI. Kohavi, Switching and Finite Automata Theory, Tata Mc Graw Hill, 3rd Edition, 2009.
2. Carl Hamacher, Zvonks Vranesic, SafeaZaky, Computer Organization, 5th Edition, Mc Graw Hill, 2001.
3. M. Morris Mano, Computer System Architecture, Third Edition, Pearson/PHI, 2017.
4. Andrew S. Tanenbaum, Structured Computer Organization, 6th Edition, PHI/Pearson, 2013.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

JAVA PROGRAMMING

Course Code: GR25A2070 **L/T/P/C: 3/0/0/3**

II Year I Semester

Course Outcomes:

1. Identify the model of Object-Oriented Programming: Abstract data types, Encapsulation,Inheritance and Polymorphism.
2. Summarize the fundamental features like Interfaces, Exceptions and Collections.
3. Correlate the advantages of Multi-threading.
4. Design interactive programs using Applets, AWT and Swings.
5. Develop real time applications using the features of Java.

UNIT I - OBJECT ORIENTED THINKING

Introduction, Need of object-oriented programming, principles of object-oriented languages, Applications of OOP, history of JAVA, Java Virtual Machine, Java features, Program structures, Installation of JDK.

Variables, Primitive data types, Identifiers- Naming Conventions, Keywords, Literals, Operators- Binary, Unary and Ternary, Expressions, Primitive Type conversion and casting, flow of control- branching, conditional, loops.

UNIT II - CLASSES, INHERITANCE, POLYMORPHISM

Classes and Objects- Classes, Objects, creating objects, methods, constructors- constructor overloading,cleaning up unused objects- Garbage collector, class variable and methods- static keyword, this keyword,arrays, Command line arguments, Nested Classes

Strings: String, String Buffer, String Tokenizer

Inheritance and Polymorphism- Types of Inheritance, deriving classes using extends keyword, super keyword, Polymorphism – Method Overloading, Method Overriding, final keyword, abstract classes.

UNIT III - INTERFACES, PACKAGES, EXCEPTIONS

Interfaces: Interface, Extending interface, interface Vs Abstract classes.

Packages- Creating Packages, using Packages, Access protection, java I/O package.

Exceptions - Introduction, Exception handling techniques-try…catch, throw, throws, finally block, userdefined Exception.

UNIT IV - MULTITHREADING, COLLECTIONS

java.lang.Thread, the main Thread, creation of new Threads, Thread priority, multithreading- using isalive() and join(), Synchronization, suspending and resuming Threads, Communication between Threads. Exploring java.io, Exploring java.util

Collections: Overview of Collection Framework: ArrayList, Vector, TreeSet, HashMap, HashTable,Iterator, Comparator.

UNIT V - APPLETS, AWT AND SWINGS

Applet class, Applet structure, an example Applet program, Applet life cycle.
Event Handling- Introduction, Event Delegation Model, Java.awt.event Description, Adapter classes,Innerclasses.
Abstract Window Toolkit: Introduction to AWT, components and containers, Button, Label, Checkbox, Radio buttons, List boxes, choice boxes, Text field and Text area, container classes, Layout Managers.
Swing: Introduction, JFrame, JApplet, JPanel, Components in swings, JList and JScroll Pane, Split Pane, JTabbed Pane, Dialog Box, Pluggable Look and feel.

Teaching methodologies:

- Power Point Presentations
- Tutorial Sheets
- Assignments

TEXTBOOKS:

1. Java: The Complete Reference, 10th edition, Herbert Schildt, Mcgraw Hill.
2. Java Fundamentals: A Comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

REFERENCE BOOKS:

1. Java for Programming, P.J. Dietel Pearson Education.
2. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.
3. Thinking in Java, Bruce Eckel, Pearson Education.
4. Programming in Java, S.Malhotra and S.Choudhary, Oxford University Press.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING ANDTECHNOLOGY

COMPUTATIONAL STATISTICS

Course Code: GR25A2084 **L/T/P/C: 3/0/0/3**
II Year I Semester

Prerequisite –Applied Engineering Mathematics, Data Structures and Algorithms

Course Outcomes:

1. Analyse multivariate data using regression, distributions, and covariance techniques.
2. Develop and validate regression models; detect outliers and co-linearity.
3. Construct discriminant functions to classify and separate data groups.
4. Perform PCA and factor analysis; interpret components and factor scores.
5. Implement clustering methods; profile, interpret, and segment data effectively.

UNIT I

Multivariate Normal Distribution: Multivariate Normal Distribution Functions, Conditional Distribution and its relation to regression model, Estimation of parameters.

Multivariate Regression: Assumptions of Multivariate Regression Models, Parameter estimation, Multivariate Analysis of variance and covariance.

UNIT II

Multiple Linear Regression Model: Standard multiple regression models with emphasis on detection of collinearity, outliers, non-normality and autocorrelation, Validation of model assumptions.

UNIT III

Discriminant Analysis: Statistical background, linear discriminant function analysis, Estimating linear discriminant functions and their properties.

UNIT IV

Principal Component Analysis: Principal components, Algorithm for conducting principal component analysis, deciding on how many principal components to retain, H-plot.
Factor Analysis: Factor analysis model, extracting common factors, determining number of factors, Transformation of factor analysis solutions, Factor scores.

UNIT V

Cluster Analysis: Introduction, Types of clustering, Correlations and distances, clustering by partitioning methods, hierarchical clustering, overlapping clustering, K-Means Clustering-Profiling and Interpreting Clusters.

TEXTBOOKS:

1. An Introduction to Multivariate Statistical Analysis, 3rd ed., T.W. Anderson, Wiley-Interscience, 2003.
2. Applied Multivariate Data Analysis, Vol I & II, 1st ed., J.D. Jobson, Springer, 1991–1992.
3. Statistical Tests for Multivariate Analysis, 1st ed., H. Kris, Springer, 1983.
4. Data Mining: Concepts and Techniques, 3rd ed., J. Han and M. Kamber, Elsevier, 2011.

REFERENCE BOOKS:

1. Regression Diagnostics, Identifying Influential Data and Sources of Collinearety, D.A. Belsey, E. Kuh and R.E. Welsch
2. Applied Linear Regression Models, J. Neter, W. Wasserman and M.H. Kutner.
3. The Foundations of Factor Analysis, A.S. Mulaik.
4. Introduction to Linear Regression Analysis, D.C. Montgomery and E.A. Peck.
5. Cluster Analysis for Applications, 1st ed., M.R. Anderberg, Academic Press, 1973.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATABASE MANAGEMENT SYSTEMS

Course Code: GR25A2071 **L/T/P/C: 3/0/0/3**

II Year I Semester

Course Outcomes:

1. Interpret the concepts of Database systems and design issues in modeling applications.
2. Develop the database using constraints and queries in SQL and PL/SQL.
3. Outline the concepts of relational model and indexing techniques.
4. Apply the Schema Refinement techniques for database design.
5. Summarize the components of transaction management in database systems.

UNIT I

Introduction to Database and System Architecture: Database Systems and their Applications, Database Vs File System, View of Data, Data Models, Database Languages - DDL and DML, Transaction Management, Database users and Administrator, Database SystemStructure.

Introduction to Database Design: ER Diagrams, Attributes, Entities and Entity sets, Relationships and Relationship sets, Extended ER Features, Conceptual Design with the ER Model, Logical database Design.

UNIT II

SQL Queries and Constraints: Types of SQL Commands, Form of Basic SQL Query, SQL Operators, Set Operators, Nested Queries, Aggregate Operators, NULL values, Keys, Integrity Constraints Over Relations, Joins, Introduction to Views, DCL Commands, Introduction to PL/SQL, Cursors, Triggers and Active Databases.

UNIT III

Relational Model: Introduction to Relational Model, Basic Structure, Database Schema, Relational Algebra, Relational Calculus.

File Organization and Indexing: Introduction, Types of File Organizations, Overview of Indexes, Types of Indexes, Index Data Structures, Tree structured Indexing, Hash based Indexing.

UNIT IV

Schema Refinement And Normal Forms: Introduction to Schema Refinement, Properties of Decomposition, Functional Dependencies, Reasoning about FD, Normal Forms – 1NF, 2NF, 3NF, BCNF, Multivalued Dependencies and 4NF.

UNIT V

Transaction Management: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, ConcurrentExecutions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability.
Concurrency Control: Lock based Protocols, Time stamp based protocols.
Recovery System: Recoveryand Atomicity, Log based recovery, Shadow Paging, Recovery with concurrent Transactions, Buffer Management.

Teaching methodologies:

- Power Point Presentations
- Tutorial Sheets
- Assignments

TEXTBOOKS:

1. "Database Management Systems",Raghurama Krishnan, Johannes Gehrke, TATA Mc GrawHill 3rd Edition.
2. "Database System Concepts", Silberschatz, Korth, Mc Grawhill, V edition.
3. "Introduction to Database Systems", C.J. Date Pearson Education.

REFERENCE BOOKS:

1. "Database Systems design, Implementation and Management", Rob &Coronel 5th Edition.
2. "Database Management Systems", P.RadhaKrishna HI-TECH Publications 2005.
3. "Database Management System", Elmasri Navate, Pearson Education.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENVIRONMENTAL SCIENCE

Course Code: GR25A2001 **L/T/P/C: 1/0/0/1**

II Year I Semester

Course Outcomes: Student will be able to

1. Understand the structure, function, and significance of ecosystems.
2. Analyze the classification, utilization, and sustainable management of natural resources, along with alternative energy options.
3. Evaluate biodiversity at genetic, species, and ecosystem levels, its values, threats, and conservation methods under national and international frameworks.
4. Identify types, sources, and impacts of environmental pollution, and apply suitable control technologies while assessing global environmental challenges and protocols.
5. Interpret environmental policies, legislation, and the EIA process to propose management plans addressing contemporary environmental and sustainability issues.

UNIT - I

Ecosystems: Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Bio magnification, ecosystem value, services and carrying capacity, Field visits.

UNIT - II

Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy resources: growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution**: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **Soil Pollution:** Sources and types, Impacts of modern agriculture, degradation of soil. Noise Pollution: Sources and Health hazards, standards, **Solid waste:** Municipal Solid Waste management, composition and characteristics of e-Waste and its management.

Pollution control technologies: Wastewater Treatment methods: Primary, secondary and Tertiary. **Global Environmental Issues and Global Efforts:** Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

UNIT - V

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition.

Slogan and Poster making on Environmental Management Plan, Contemporary Environmental Issues (Climate change – Impact on air, water, biological and Socioeconomical aspects); Sustainable development goals (SDGs); Global environmental challenges; Environmental policies.

TEXTBOOKS:

1. Introduction to Environmental Science by Y. Anjaneyulu, BS. Publications.
2. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University
   Grants Commission.
3. Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

JAVA PROGRAMMING LAB

Course Code: GR25A2072 **L/T/P/C: 0/0/4/2**

II Year I Semester

Course Outcomes:

1. Analyze a problem, identify and define the computing requirements appropriate to its solution using object-oriented programming concepts.
2. Design the applications using Inheritance, Polymorphism and Synchronization concepts.
3. Illustrate exception handling at Compile time and Run time.
4. Solve the real-world problems using Java Collection framework.
5. Develop GUI applications using Applets, AWT and Swings.

TASK 1

Write java programs that implement the following

a) Class and object
b) Constructor
c) Parameterized constructor
d) Method overloading
e) Constructor overloading.

TASK 2

a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is apalindrome.
b) Write a Java program for sorting a given list of names in ascending order.
c) Write a Java Program that reads a line of integers, and then displays each integer and the sum of allthe integers (Use StringTokenizer class of java.util)

TASK 3

Write java programs that uses the following keywords

a) this b)super c)static d)final

TASK 4

a) Write a java programto implement method overriding
b) Write a java program to implement dynamic method dispatch.
c) Write a Java program to implement multiple inheritance.
d) Write a java program that uses access specifiers.

TASK 5

a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of thefile in bytes.

b) Write a Java program that reads a file and displays the file on the screen, with a line number before eachline.
c) Write a Java program that displays the number of characters, lines and words in a text file

TASK 6

a) Write a Java program for handling Checked Exceptions.
b) Write a Java program for handling Unchecked Exceptions.

TASK 7

a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.
b) Write a Java program that correctly implements producer consumer problem using the concept of interthread communication.

TASK 8

Write a program illustrating following collections framework

a) ArrayList b) Vector c) HashTable d) Stack

TASK 9

a) Develop an applet that displays a simple message.
b) Develop an applet that receives an integer in one text field and compute its factorial value and return itin another text field, when the button named "Compute" is clicked.
c) Write a Java programthat works as a simple calculator. Use a grid layout to arrange button for the digitsand for the +, -,*, % operations. Add a text field to display the result.

TASK 10

a) Write a Java program for handling mouse events.
b) Write a Java program for handling key events.

TASK 11

a) Write a program that creates a user interface to perform integer divisions. The user enters two numbersin the text fields Num1 and Num 2.
b) The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1or Num2 were not an integer, the program would throw Number Format Exception. If Num2 were Zero, the program would throw an

Arithmetic Exception and display the exception in a message dialogbox.

TASK 12

a) Write a java program that simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.

b) Write a Java program that allows the user to draw lines, rectangles and ovals.

c) Create a table in Table.txt file such that the first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to displaythe table using JTable component.

TEXTBOOKS:

1. Java: The Complete Reference, 10th edition, Herbert Schildt, Mcgraw Hill.
2. Java Fundamentals - A Comprehensive introduction, Herbert schildt and Dale skrien, TMH.

REFERENCE BOOKS:

1. Java for programming, P.J.Dietel Pearson education (OR) Java: How to Program P.J.Dietel andH.M.Dietel,PHI
2. Object Oriented Programming through java, P.Radha Krishna, Universities Press.
3. Thinking in Java, Bruce Eckel, Pearson Education
4. Programming in Java, S.Malhotra and S.Choudhary, Oxford University Press.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTATIONAL STATISTICS LAB

Course Code: GR25A2085 **L/T/P/C: 0/0/2/1**
II Year I Semester

Prerequisite – Python Programming, Applied Engineering Mathematics, Data Structures and Algorithms

Course Outcomes:

1. Understanding the basics utilization of python functions in real time operational processing include Data Structures, Flow control, Functions, and Files.
2. To implement and visualize statistical models using Python libraries.
3. Apply to perform correlation, regression, and time series forecasting.
4. Analysis the multivariate statistical methods like LDA, PCA, factor analysis, and clustering.
5. Create a model to integrate data pre-processing, modelling, and visualization in a unified workflow

LIST OF EXPERIMENTS:

TASK-1 (Control Flow)
a) **Experiment:** Write a program to check whether the given number is even or not.
Real Example: Determining even/odd IDs in a database.
Use Case: Useful in load balancing, ID allocation.
b) **Experiment:** Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, ... 1/10.
Real Example: Currency conversion fractions.
Use Case: Helpful in financial calculations.

c) **Experiment:** Write a program using a while loop that asks the user for a number and prints a countdown from that number to zero.
Real Example: Countdown timer in events.
Use Case: Useful in games, event management systems

TASK-2 (Functions)
a) **Experiment:** Write a Python program to swap given numbers using Functions.
Real Example: Swapping temporary variables in simulations.
Use Case: Memory management, algorithms.
b) **Experiment:** Write a Python program to find Fibonacci numbers using Recursive function.
Real Example: Financial growth prediction models.
Use Case: Recurrence-based systems, population growth modeling.

TASK-3 (Data Structures)

a) **Experiment:** Write a program to count the number of characters in a string and store them in a dictionary.

Real Example: Word frequency in search engines.

Use Case: Text mining, NLP.

b) **Experiment:** Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Real Example: Contact book storing birthdays.

Use Case: Reminder systems, calendar apps.

TASK-4 (Files)

a) **Experiment:** Write a script named copyfile.py to copy contents of one text file into another.

Real Example: Backup systems.

Use Case: Data migration.

b) **Experiment:** Write a program that inputs a text file and prints all unique words in alphabetical order.

Real Example: Indexing system in libraries.

Use Case: Search engines, plagiarism detection.

TASK-5 (Matplotlib Basics)

a) **Experiment:** Import Iris dataset (UCI) and Wine Reviews dataset (Kaggle).

Use Case: Dataset exploration.

b) **Experiment:** Scatter plot Sepal Length vs Sepal Width.

Real Example: Plant classification.

c) **Experiment:** Line chart plotting each column in dataset.

Real Example: Stock market analysis.

d) **Experiment:** Histogram and Bar chart for Wine Reviews scores.

Real Example: Customer ratings.

TASK-6 (Matplotlib Advanced)

a) **Experiment:** Using text() command add text to axes of figures.

Use Case: Annotated charts.

b) **Experiment:** Using annotate() label parts of axes in figures.

Use Case: Highlighting trends.

c) **Experiment:** Using Locator and Formatter objects set axis properties.

Use Case: Custom ticks in plots.

TASK-7 (Matplotlib Patches)

a) **Experiment:** Draw a rectangle patch to a plot.

Use Case: Highlighting zones in graphs.

b) **Experiment:** Draw a circular patch at a given center with a given radius.

Use Case: Scatter highlights.

TASK-8 (Advanced Functions)

a) **Experiment:** Demonstrate use of setp() and getp() methods.

Use Case: Dynamic property changes.

b) **Experiment:** Write a Python program to implement Multiple Regression.
Real Example: Predicting house prices.
Use Case: Machine learning models.

TASK-9 (Multivariate Analysis)

a) **Experiment:** Read Multivariate Data from Wine dataset.
Use Case: Feature exploration.
b) **Experiment:** Plot Multivariate Data and calculate summary statistics.
Real Example: Customer segmentation.

TASK-10 (PCA + Classification)

a) **Experiment:** Read the Iris dataset.
b) **Experiment:** Apply Principal Component Analysis (PCA) for dimensionality reduction.
Use Case: Data simplification.
c) **Experiment:** Classify data using Random Forest Classifier.
Real Example: Species classification.
d) **Experiment:** Evaluate performance of the model.
Use Case: Accuracy measurement.

TASK-11 (LDA + Classification)

a) **Experiment:** Read the Iris dataset.
b) **Experiment:** Perform Linear Discriminant Analysis (LDA).
c) **Experiment:** Classify using Random Forest Classifier.
d) **Experiment:** Evaluate model performance.
Comparison: Compare LDA performance with PCA from TASK-10.

TASK-12 (Cluster Analysis using K-Means)

a) **Experiment:** Read Titanic dataset (UCI).
b) **Experiment:** Apply Data Preprocessing techniques.
c) **Experiment:** Use PCA for dimensionality reduction.
d) **Experiment:** Perform K-Means clustering.
Real Example: Passenger survival grouping.
Use Case: Market segmentation, customer clustering.

TEXTBOOKS

1. Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis* (3rd ed.). Wiley.
2. Jobson, J. D. (1991). *Applied Multivariate Data Analysis, Volume I: Regression and Experimental Design*; Volume II: *Categorical and Multivariate Methods*. Springer.
3. Kris, H. (1982). *Statistical Tests for Multivariate Analysis*. Wiley.
4. Lutz, M. (2010). *Programming Python* (4th ed.). O'Reilly Media.
5. Hall, T., & Stacey, J.-P. (2009). *Python 3 for Absolute Beginners*. Apress.

6. Hetland, M. L. (2005). *Beginning Python: From Novice to Professional*. Apress.

REFERENCE BOOKS

1. Belsey, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley.
2. Neter, J., Wasserman, W., & Kutner, M. H. (1990). *Applied Linear Regression Models* (3rd ed.). Irwin.
3. Mulaik, S. A. (2009). *Foundations of Factor Analysis* (2nd ed.). CRC Press.
4. Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). Wiley.
5. Anderberg, M. R. (2014). *Cluster Analysis for Applications* (SIAM Reprint ed.). SIAM.
6. Morrison, D. F. (2004). *Multivariate Statistical Methods* (4th ed.). Duxbury Press.
7. McKinney, W. (2022). *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O’Reilly Media.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATABASE MANAGEMENT SYSTEMS LAB

Course Code: GR25A2073 **L/T/P/C: 0/0/3/1.5**

II Year I Semester

Course Outcomes:

1. Demonstrate the use of DDL and DML commands in SQL.
2. Apply the basic SELECT operations for data retrieval.
3. Illustrate the SQL concepts to retrieve data from multiple tables.
4. Construct PL/SQL code to work with database objects.
5. Experiment with procedural constructs and exception handling to develop applications in the database systems.

TASK 1 : DDL commands (Create, Alter, Drop, Truncate)

1. Create a table EMP with the following structure:

Name	**Data Type**
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
MGRID	NUMBER(6)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

2. Add a column **commission** to the emp table. Commission should be numeric with null valuesallowed.
3. Alter the **job** field of EMP table by modifying its size.
4. Create a table DEPT with the following structure:

Name	**DataType**
DEPTNO	NUMBER(3)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

DEPTNO as the primary key

5. Add constraints to the EMP table with **empno** as the primary key and **deptno** as the foreign key referencing the DEPT table.
6. Add check constraint to the EMP table to check the **empno** value with **empno**>100.
7. Add NOT NULL constraint on **sal** field with default value 5000, otherwise it should accept the values from the user.
8. Add columns **dob, doj** with date data type to the EMP table. Drop the column **doj** from the EMP table.
9. Create EMP1, EMP2 tables as copy of EMP table. Drop EMP1 table and truncate EMP2 table.

TASK 2: DML COMMANDS (Insert, Update, Delete)

1. Insert 5 records into DEPT table.
2. Insert 11 records into EMP table.
3. Update the EMP table to set the value of **commission** of all employees to Rs1000/- who are working as “clerk”.
4. Delete the records from EMP table whose **job** is “Admin”.
5. Delete the rows from DEPT table whose **deptno** is 10.

TASK 3: DQL COMMAND (SELECT) - SQL Operators and Order by Clause

Note: Use EMPLOYEES and DEPARTMENTS tables of HR Schema

1. List the records in the EMPLOYEES table by sorting the salary in descending order.
2. Displayonly those employees whose department number is 30.
3. Displaythe unique department numbers from EMPLOYEES table.
4. List the employee name, salaryand 15% rise in salary. Label the column as “pay_hike”.
5. Display the rows whose salary ranges from 5000 to 7500.
6. Display all the employees in department 10 or 20 in alphabetical order of employee names.
7. List the employee names who do not earn commission.
8. Display all the details of the employees with 5-character names with ‘S’ as starting character.
9. Display joining date of all employees in the year of 1998.
10. List out the employee names whose salary is greater than 5000 and less than 6000.

TASK 4: SQL Aggregate Functions, Group By clause, Having clause

1. Count the total records in the EMPLOYEES table.
2. Calculate the total and average salary of the employees.
3. Determine the max and min salary and rename the column as “max_salary” and “min_salary”.
4. Find number of uniquedepartments from the EMPLOYEES table.
5. Display job wise sum, average, maximum, and minimum salary from EMPLOYEES table.
6. Display maximum salaries of all the departments having maximum salary >2000.
7. Display job wise sum, average, maximum, minimum salaries in department 10 having average salary greater than1000. Sort the result with the sum of salary in descending order.

TASK 5: SQL Functions

1. Display the employee name concatenated with employee number.
2. Display the employee name with half of employee name in upper case and half in lowercase.
3. Display the month name of “14-OCT-09” in full.
4. Display the date of joining of all employees in the format “dd-mm-yy”.
5. Display the date after two months of hire date of employees.
6. Display the last date of the month in “05-OCT-09”.
7. Display the hire date by rounding the date with respect to month and year.
8. Display the commission earned by employees. If they do not earn commission,

display it as “NoCommission”.

TASK 6: Nested Queries

1. Display the salary of the third highest paid employee in EMPLOYEES table.
2. Display the employee name and salary of employees whose salary is greater than the minimum salary of the company and job title starts with ‘I’.
3. Write a queryto display information about employees who earn more than any employee in department number 30.
4. Display the employees who have the same job as “Jones” and whose salary is greater than or equalto the salary of “Ford”.
5. List out the employee names who get the salarygreater than the maximum salary of departments with department number 20, 30.
6. Display the maximum salary of the departments where maximum salary is greater than 9000.
7. Create a table employee with the same structure as EMPLOYEES table and insert rows into the table using select clause.
8. Create MANAGER table from the EMPLOYEES table which should hold details only about the managers.

TASK 7: Joins, Set Operators.

1. Display all the EMPLOYEES and the DEPARTMENTS information implementing a left outer join.
2. Display the employee name and department name in which they are working implementing a fullouter join.
3. Write a queryto display the employee name, salary and their manager’s name for everyemployee.
4. Write a queryto display the employee name, job, employee number, department name and location for each department, even if there are no employees.
5. Display the details of employees those who draw the same salary.
6. Display the names of employees who did not change their job at least once. (Use Set Operators)
7. Display the names of employees whose current job_id is same as their previous one. (Use Set Operators)
8. Display the names of employees with their current and previous job details. (Use Set Operators)

TASK 8: Views

1. Create a view that displays the employee id, name and salary of employees who belong to10th department using with check option.
2. Create a view with read only option that displays the employee name and their department name.
3. Display all the views generated.
4. Execute the DML commands on the views created.

TASK 9: Sequence and Index

1. Write a PL/SQL codeto retrieve the employee name, hire date and designation of an

employee whosenumber is given as input bythe user.

2. Write a PL/SQL code to calculate tax of employee.
3. Write a PL/SQL programto display top ten employee details based on salary using cursors.
4. Write a PL/SQL program to update the commission values for all the employees with salary lessthan 2000, by adding 1000 to the existing values.

TASK 10: TCL COMMANDS (Save Point, Rollback, Commit)

TASK 11: Triggers, Procedures, and Functions

1. Write a trigger on employee table that shows the old and new values of employee name after updatingon employee name.
2. Write a PL/SQL procedure to insert, delete, and update the records in the EMPLOYEES table.
3. Write a PL/SQL function that accepts the department number and returns the total salary of thatdepartment.

TASK 12: Exceptions and Packages

1. Write PL/SQL program to handle predefined exceptions.
2. Write PL/SQL program to handle user defined exception.
3. Write a PL/SQL code to create
   a) Package specification
   b) Package bodyto insert, update, delete and retrieve data on EMPLOYEES table.

TEXTBOOKS

1. The Complete Reference, 3rd edition by James R.Groff, Paul N.Weinberg, Andrew J. Oppel
2. SQL & PL/SQL for Oracle10g, Black Book, Dr. P.S.Deshpande.

REFERENCE BOOKS

1. Database Systems design, Implementation and Management", Rob &Coronel 5th Edition.
2. "Database Management Systems", P.RadhaKrishna HI-TECH Publications 2005.
3. "Database Management System", Elmasri Navate, Pearson Education.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA VISUALIZATION LAB

Course Code: GR25A2097 **L/T/P/C: 0/0/3/1.5**

II Year I Semester

Prerequisites:

Students should have prior knowledge of

1. Basic programming skills.
2. Basic Probability and Statistics.

Course Outcomes:

On successful completion of this course, students will be able to

1. Understand and apply fundamental data visualization workflows and design principles, including perceptual rules, storytelling techniques, and chart selection heuristics, to effectively represent structured, semi-structured, and unstructured data for diverse business and real-world applications.
2. Create and analyse interactive visualizations and dashboards.
3. Perform and automate data cleaning, transformation, joining, and aggregation using appropriate data preparation and ETL techniques, ensuring high-quality, consistent, and integrated datasets for accurate analysis and visualization.
4. Apply advanced visualization techniques such as 2D and 3D plots, manifold learning-based visualization, network/graph-based analytics, and geospatial mapping to explore high-dimensional data, uncover hidden patterns, and support decision-making.
5. Interpret and communicate insights related to product, customer, and order-level performance through interactive dashboards and business intelligence reports, enabling organizations to make data-driven strategic and operational decisions.

Reference Datasets for implementation:

a) Global Superstore dataset (in-build)

b) Students Performance in Exams (from Kaggle)

c) Tourism in India Dataset (from Kaggle)

TASK 1: Data Visualization Workflow

- Experiment: Demonstrate the data visualization workflow by defining visualization concepts, describing processes (data acquisition → cleaning → exploration → visualization → interpretation).
- Real Example: Sales Dashboard Preparation
    - Understand data (e.g., Global Superstore dataset).
    - Identify KPIs (Sales, Profit, Quantity).
    - Choose proper chart types.
- Use Case: Used by business analysts for decision-making.

Sub-Task:

- Develop a Bar Chart: Visualize and analyze sales performance across product categories using a given sales dataset.
- Use Case: Identify top-performing product categories.

TASK 2: Data Representation Experiments

- Query: Which Category contributes the most to Sales?
- Experiments:
  - Categorical Chart: Bar/Pie Chart – Show sales by category.
  - Hierarchical Chart: TreeMap/Stacked Bar – Sales by Category → Sub-Category.
  - Relational Chart: Scatter/Bubble – Sales vs Profit highlighting category.
  - Temporal Chart: Line/Bump Chart – Sales trend over time by category.
  - Spatial Chart: Filled/Symbol Map – Sales by category across regions.
- Real Example: Market analysis for sales contribution.
- Use Case: Helps stakeholders understand contributions across dimensions.

TASK 3: 2-D Data Representation

- Experiments:

  a) Bar chart – total sales by category, identify highest contributor.

  b) Clustered bar – compare sales by category segmented by region.

  c) Dot plot – find category with lowest sales.

  d) Connected dot plot – analyze profit differences across regions.

  e) Pictogram – represent sales volume per category; highlight top seller.

  f) Proportional shape – visualize sales contribution per category.

  g) Bubble chart – explore revenue, profit, and sales relationships.

  h) Radar chart – evaluate balance among revenue, profit, and sales.

  i) Polar chart – visualize revenue distribution across categories.
- Real Example: E-commerce site analyzing product performance.
- Use Case: Helps visualize relationships and patterns for strategy.

TASK 4: Advanced 2-D Data Visualization

- Experiments:

  a) Range chart – variation in prices; find widest range.

  b) Box & whisker – compare distribution; detect outliers.

  c) Univariate scatter – relationship between satisfaction & sales.

  d) Histogram – order quantity distribution; find most common.

  e) Word cloud – analyze customer feedback themes.

  f) Pie chart – sales distribution by region; find top region.

  g) Waffle chart – sales proportion by category.

  h) Stacked bar – contribution of each region to total sales.

  i) Back-to-back bar – compare sales of two products.

  j) Treemap – hierarchical structure of sales; top sub-category.

  k) Scatter – product price vs ratings; identify patterns.

  l) Line chart – monthly sales trend; highest/lowest months.

  m) Dashboard – combine multiple visualizations for overall performance.
- Real Example: Retail chain analyzing sales & customer feedback.
- Use Case: Supports deep analysis and actionable insights.

TASK 5: Data Cleaning & Transformation

- Experiment: Remove nulls & duplicates, standardize dates, rename columns, fix data types.
- Real Example: Cleaning messy sales data for reporting.
- Use Case: Ensures accurate, consistent analytics.

TASK 6: Joining, Aggregation & Output

- Experiment:
  - Join Orders with Returns table.
  - Aggregate by Category & Region → Total Sales & Profit.
  - Create calculated fields (e.g., Profit Ratio = Profit/Sales).
  - Output cleaned dataset.
- Real Example: E-commerce return impact analysis.
- Use Case: Data preparation for business reports.

TASK 7 – TASK 11: 3D & Advanced Data Visualization (Python)

TASK 7: Implement Surface Plots, Contour Plots, Hidden Surface Removal, PM3D Coloring

- Experiment:
  - Load a 3D dataset (e.g., elevation data or sales data across multiple regions and time).
  - Generate surface and contour plots using tools like Matplotlib (Python) or Gnuplot.
  - Implement hidden surface removal (so only visible surfaces are shown).
  - Apply PM3D coloring to indicate intensity (e.g., heat levels or sales density).
- Real Example:
  - Elevation maps for different terrains using geospatial datasets (like USGS or DEM data).
- Use Case:
  - 3D mapping in geospatial analytics – visualizing terrain elevation, rainfall distribution, or population density.

TASK 8: Multi-Dimensional Data Visualization for Relationships Across Variables

- Experiment:
  - Use a dataset with multiple numerical features (e.g., marketing dataset: ad spend, sales, ROI, customer reach).
  - Create multi-variable visualizations such as bubble charts, parallel coordinates plots, or pair plots (Seaborn).
  - Analyze how one variable affects another (e.g., does higher ad spend increase ROI?).
- Real Example:
  - Visualizing sales performance vs marketing spend across different channels (TV, social media, email campaigns).
- Use Case:
  - Marketing analysis – identifying which channels give the best ROI and optimize ad spend.

TASK 9: Manifold Visualization (e.g., t-SNE) to Explore High-Dimensional Datasets

- Experiment:
  - Take a high-dimensional dataset (e.g., customer purchase data with 50+ features).
  - Apply dimensionality reduction using t-SNE or PCA to project it into 2D/3D space.
  - Plot the clusters and interpret customer segments.
- Real Example:
  - Retail customer segmentation based on buying patterns (e.g., RFM analysis – Recency, Frequency, Monetary value).
- Use Case:

- Customer segmentation – grouping customers with similar purchasing behavior for targeted marketing.

TASK 10: Graph Data Visualization – Analyze & Display Network Structures

- Experiment:
  - Use a graph/network dataset (e.g., social network followers, website hyperlink structure).
  - Create visualizations using NetworkX (Python) or Gephi to display nodes and edges.
  - Analyze metrics like centrality, shortest paths, and communities.
- Real Example:
  - Visualizing a Facebook/Twitter network showing relationships between users and influencers.
- Use Case:
  - Social network analysis – detect influencers, communities, and information flow.

TASK 11: Annotation Techniques – Enhance Clarity and Interpretation

- Experiment:
  - Take any visualization (bar, line, scatter).
  - Add annotations, labels, reference lines, tooltips to highlight key metrics (e.g., highest sales point, anomaly in data).
  - Compare the chart before and after annotation to see clarity improvements.
- Real Example:
  - Annotating quarterly revenue growth and highlighting dips/spikes in a sales report dashboard.
- Use Case:
  - Presenting KPIs effectively to stakeholders – making dashboards understandable for decision-makers.

TASK 12: Global Superstore Advanced Analysis

- Query: Identify Sub-Categories Where Top 20 Customers have higher sales than others in the same region.
- Further Analysis:
  1. Product level analysis.
  2. Customer level analysis.
  3. Order level analysis.
- Real Example: Top customer segmentation and targeted marketing.
- Use Case: Identifies loyal customers & informs business strategies.

Textbooks:

1. Andy Kirk, Data Visualization: A Handbook for Data Driven Design, 3rd Edition Paperback, 2025.
2. Cole Nussbaumer Knaflic, Storytelling with Data: A Data Visualization Guide for Business Professionals, Wiley, 2015.
3. Philipp K. Janert, Gnuplot in Action, 2nd Edition, 2016.

Reference Books:

1. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 3rd Edition, 2022.

2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 2nd Edition, 2022.
3. Kyran Dale, Data Visualization with Python and JavaScript, 2nd Edition, 2017.

II YEAR
II SEMESTER

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

OPERATING SYSTEMS CONCEPTS

Course Code: GR25A2086 **L/T/P/C: 3/0/0/3**

II Year II Semester

Prerequisite: Fundamental concepts of data structures

Course Outcomes:

1. Understand the fundamental concepts, architecture, and functionalities of operating systems.
2. Analyze and implement process management techniques including scheduling, synchronization, and inter-process communication.
3. Identify, analyze, and apply deadlock handling techniques for safe resource allocation.
4. Understand memory management strategies including paging, segmentation, and page replacement algorithms.
5. Apply concepts of file systems, storage management, and disk scheduling to design efficient storage solutions. Integrate theoretical OS concepts with practical problem- solving using system-level programming in C/Unix.

UNIT I

Introduction: Concept of Operating Systems (OS), Generations of OS, Types of OS, OS Services, Interrupt handling and System Calls, Basic architectural concepts of an OS, Concept of Virtual Machine, Resource Manager view, process view and hierarchical view of an OS.
Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

UNIT II

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
Scheduling algorithms: Pre-emptive and non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.

UNIT III

Inter-process Communication: Concurrent processes, precedence graphs, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Semaphores, Strict Alternation, Peterson’s Solution, The Producer / Consumer Problem, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader’s & Writer Problem, Dining Philosopher Problem, Barber’s shop problem.
Concurrent Programming: Critical region, conditional critical region, monitors, concurrent languages, communicating sequential process (CSP); Deadlocks - prevention, avoidance, detection and recovery.

UNIT IV

Memory Management: Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page allocation, Partitioning, Paging, Page fault, Working Set, Segmentation, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

UNIT V

I/O Hardware: I/O devices, Device controllers, Direct Memory Access, Principles of I/O. **File Management**: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation(linear list, hash table), efficiency and performance.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

Case study: UNIX OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX system calls.

TEXTBOOKS:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating system concepts (10th ed.). Wiley.
2. Stallings, W. (2021). Operating systems: Internals and design principles (9th ed.). Pearson.

REFERENCE BOOKS:

1. Tanenbaum, A. S., & Bos, H. (2015). Modern operating systems (4th ed.). Pearson.
2. Crowley, C. P. (1997). Operating systems: A design-oriented approach. McGraw-Hill.
3. Nutt, G. J. (2004). Operating systems: A modern perspective (3rd ed.). Addison-Wesley.
4. Bach, M. J. (1986). The design of the UNIX operating system. Prentice Hall.
5. Bovet, D. P., & Cesati, M. (2005). Understanding the Linux kernel (3rd ed.). O’Reilly Media.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

PRINCIPLES OF SOFTWARE ENGINEERING

Course Code: GR25A2087 **L/T/P/C: 3/0/0/3**
II Year II Semester

Prerequisite: Basic knowledge of programming for problem solving

Course Outcomes:

1. Understand principles and processes for developing high-quality software systems.
2. Apply project management techniques for planning, estimating, and controlling software projects.
3. Evaluate software quality models and assess reliability to ensure robust systems.
4. Analyze user requirements and design solutions using modeling and domain knowledge frameworks.
5. Perform verification, validation, and testing to ensure the correctness and performance of software.

UNIT I
Introduction: Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; engineering approach to software development; role of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline.

UNIT II
Software Project Management: Basic concepts of life cycle models – different models and milestones; software project planning – identification of activities and resources; concepts of feasibility study; techniques for estimation of schedule and effort; software cost estimation models and concepts of software engineering economics; techniques of software project control and reporting; introduction to measurement of software size; introduction to the concepts of risk and its mitigation; configuration management.

UNIT III
Software Quality and Reliability: Internal and external qualities; process and product quality; principles to achieve software quality; introduction to different software quality models like McCall, Boehm, FURPS / FURPS+, Dromey, ISO – 9126; introduction to Capability Maturity Models (CMM and CMMI); introduction to software reliability, reliability models and estimation.

UNIT IV
Problem Space Understanding:
How an industry works, how an IT company works, How IT supports business, Problem Space Understanding, Knowledge Driven Development (KDD), Domain knowledge framework of KDD,usage of domain knowledge framework in Insurance, Banking and Automobile, KDD as a project delivery methodology, Linking domain knowledge to software development, An example to illustrate this, A case study to produce a KDD artifact using Agile.
Software Requirements Analysis, Design and Construction: Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modeling – decision tables, event tables, state transition tables, Petri nets; requirements documentation through use cases; introduction to UML, introduction to software metrics and metrics-based control methods; measures of code and design quality.

UNIT V

Software Testing: Introduction to faults and failures; basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests – equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction- based testing; testing for non-functional requirements – volume, performance and efficiency; concepts of inspection.

TEXTBOOKS:

1. Software Engineering, 10th ed., I. Sommerville, Pearson, 2015.
2. Fundamentals of Software Engineering, 2nd ed., C. Ghezzi, M. Jazayeri, and D. Mandrioli,
   Pearson, 2002.

REFERENCE BOOKS:

1. Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices, M. Jackson, Addison-Wesley, 1995.
2. The Unified Software Development Process, I. Jacobson, G. Booch, and J. Rumbaugh, Addison-Wesley, 1999.
3. Design Patterns: Elements of Reusable Object-Oriented Software, E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Addison-Wesley, 1994.
4. Software Metrics: A Rigorous and Practical Approach, N.E. Fenton and S.L. Pfleeger, PWS Publishing, 1997.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ALGORITHM DESIGN AND ANALYSIS

Course Code: GR25A2088 **L/T/P/C :3/0/0/3**

II Year II Semester

Prerequisite: Students are expected to have prior knowledge of programming fundamentals, data structures, and discrete mathematics**.**

Course Outcomes:

1. Analyze the performance of algorithms and represent using asymptotic notations.
2. Differentiate and demonstrate various algorithm design strategies.
3. Solve various problems using algorithmic design paradigms and can analyze their complexities.
4. Demonstrate and solve the tree traversal problems and analyze its complexity.
5. Distinguish NP complete and NP hard problems.

UNIT I

Introduction: Characteristics of Algorithms. Analysis of Algorithm: Asymptotic analysis of Complexity Bounds – Best, Average and Worst-Case behavior; Performance Measurements of Algorithm, Time and Space Trade-Offs, Analysis of Recursive Algorithms through Recurrence Relations: Substitution Method, Recursion Tree Method and Masters' Theorem.

UNIT II

Fundamental Algorithmic Strategies: Brute-Force, Heuristics, Greedy, Dynamic Programming, Branch and Bound and Backtracking methodologies; Illustrations of these techniques for Problem-Solving, Bin Packing, Knapsack, Travelling Salesman Problem.

UNIT III

Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

UNIT IV

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

UNIT V

Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE, Introduction to Quantum Algorithms.

TEXTBOOKS:

1. Fundamentals of Computer Algorithms, E. Horowitz and S. Sahni.
2. The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft and J. Ullman.

REFERENCE BOOKS:

1. "Introduction to Algorithms", T. H. Cormen, C. E. Leiserson and R. L. Rivest.
2. "Computer Algorithms: Introduction to Design and Analysis", S. Baase.
3. "The Art of Computer Programming", Vol. 1, Vol. 2 and Vol. 3, .D. E. Knuth.
4. "Quantum Computation and Quantum Information" Michael A. Nielsen and Isaac L. Chuang.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
OPERATIONAL RESEARCH

Course Code: GR25A2089 **L/T/P/C : 3/0/0/3**

II Year II Semester

Course Outcomes:

1. To impart knowledge in concepts, tools of operations research and to understand and apply the theoretical workings method for linear programming and apply various linear programming techniques for optimal allocation of limited resources.
2. To be able to build and solve transportation and assignment problems using appropriate method
3. To be exceptional to design and solve simple models of project scheduling techniques such as PERT & CPM in developing critical thinking and objective analysis of decision problems.
4. To understand the inventory management elements including the relevant related costs and distinguish various inventory models for developing proper inventory control policies.
5. To examine situations in which queuing problems are generated and appreciate simulation methodology.

UNIT- I

Introduction to OR: Origin of OR and its definition. Concept of optimizing performance measure, Types of OR problems, Deterministic vs. Stochastic optimization, Phases of OR problem approach – problem formulation, building mathematical model, deriving solutions, validating model, controlling, and implementing solution.

Linear Programming: Linear programming – Examples from industrial cases, formulation & definitions, Matrix form. Implicit assumptions of LPP.

Some basic concepts and results of linear algebra – Vectors, Matrices, Linear Independence / Dependence of vectors, Rank, Basis, System of linear eqns., Hyperplane, Convex set, Convex polyhedron, Extreme points, Basic feasible solutions.

Geometric method: 2-variable case, Special cases – infeasibility, unboundedness, redundancy & degeneracy, Sensitivity analysis.

Simplex Algorithm – slack, surplus & artificial variables, computational details, big-M method, identification and resolution of special cases through simplex iterations.

Duality – formulation, results, fundamental theorem of duality, dual-simplex and primal-dual algorithms.

UNIT - II

Transportation and Assignment problems: TP - Examples, Definitions – decision variables, supply & demand constraints, formulation, Balanced & unbalanced situations, Solution methods – NWCR, minimum cost and VAM, test for optimality (MODI method), degeneracy and its resolution.

AP - Examples, Definitions – decision variables, constraints, formulation, Balanced & unbalanced situations, Solution method – Hungarian, test for optimality (MODI method), degeneracy & its resolution.

UNIT- III

PERT – CPM: Project definition, Project scheduling techniques – Gantt chart, PERT & CPM, Determination of critical paths, Estimation of Project time and its variance in PERT using statistical principles, Concept of project crashing/time-cost trade-off.

UNIT -IV

Inventory Control: Functions of inventory and its disadvantages, ABC analysis, Concept of inventory costs, Basics of inventory policy (order, lead time, types), Fixed order-quantity models – EOQ, POQ & Quantity discount models. EOQ models for discrete units, sensitivity analysis and Robustness, Special cases of EOQ models for safety stock with known / unknown stock out situations, models under prescribed policy, Probabilistic situations.

UNIT V

Queuing Theory: Definitions – queue (waiting line), waiting costs, characteristics (arrival, queue, service discipline) of queuing system, queue types (channel vs. phase).

Kendall’s notation, Little’s law, steady state behaviour, Poisson’s Process & queue, Models with examples - M/M/1 and its performance measures; M/M/m and its performance measures; brief description about some special models.

Simulation Methodology: Definition and steps of simulation, random number, random number generator, Discrete Event System Simulation – clock, event list, Application in Scheduling, Queuing systems and Inventory systems.

TEXTBOOKS:

1. Operations Research: An Introduction. H.A. Taha.

REFERENCE BOOKS:

1. Linear Programming. K.G. Murthy.
2. Linear Programming. G. Hadley.
3. Principles of OR with Application to Managerial Decisions. H.M. Wagner.
4. Introduction to Operations Research. F.S. Hiller and G.J. Lieberman.
5. Elements of Queuing Theory. Thomas L. Saaty.
6. Operations Research and Management Science, Handbook: Edited By A. Ravi Ravindran.
7. Management Guide to PERT/CPM. Wiest & Levy.
8. Modern Inventory Management. J.W. Prichard and R.H. Eagle.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

MATHEMATICAL AND STATISTICAL FOUNDATIONS

Course Code: GR25A2006 **L/T/P/C : 3/0/0/3**

II Year II Semester

Pre-requisites : Mathematics courses of first year of study.

Course outcomes: After learning the contents of this course, the student must be able to

1. Apply the number theory concepts to cryptography and computer algorithms.
2. Apply the concepts of probability and distributions to some case studies.
3. Apply the concepts of normal distribution, sampling distributions and analyze the role of Central Limit Theorem.
4. Apply inferential statistical methods to interpret and explain phenomena using both large and small samples.
5. Analyze bivariate data using statistical modelling.

UNIT-I : Basics of Number Theory

Greatest Common Divisors and Prime Factorization: Greatest common divisors – The Euclidean algorithm – The fundamental theorem of arithmetic – Factorization of integers and the Fermat numbers. Congruences: Introduction to congruences – Linear congruences.

UNIT-II : Random Variables and Probability Distributions

Concept of a Random Variable – Discrete Probability Distributions – Continuous Probability Distributions – Mean, Variance and Moments about mean of a Random Variable

Theoretical Discrete Probability Distributions: Binomial Distribution – Poisson distribution

UNIT-III : Continuous Distributions and Sampling

Normal Distribution – Areas under the Normal Curve – Applications of the Normal Distribution – Normal Approximation to the Binomial Distributions.

Fundamental Sampling Distributions : Random Sampling – Some Important Statistics (Sample mean and Proportion) – Sampling Distributions – Sampling Distribution of Means – Central Limit Theorem.

UNIT-IV : Tests of Hypotheses (Large and Small Samples)

Statistical Hypotheses: General Concepts – Testing a Statistical Hypothesis. Single sample : Tests concerning a single mean. Two samples : Tests on two mean (Unknown for equal variance). One sample: Test on a single proportion . Two samples : Tests on two proportions. Two- sample tests concerning variances: F-distribution

UNIT-V : Applied Statistics

Curve fitting by the method of least squares – Fitting of straight lines – Second degree parabolas, exponential and power curves – Correlation (Karl Pearson and Spearman) and Regression of two variables

TEXT BOOKS :

1. Kenneth H. Rosen, Elementary Number Theory & its Applications, sixth edition, Addison Wesley, ISBN 978 0-321-50031-1.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical Statistics, Khanna publications.

REFERENCE BOOKS :

1. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons, Ltd, 2004.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

INDIAN KNOWLEDGE SYSTEM

Course Code: GR25A2003 **L/T/P/C: 1/ 0/ 0/1**

II Year II Semester

Course Outcomes: Students will be able to

1. Understand and analyze the core concepts of Bharatiya Jnana Parampara and foundational texts like Vedas and Upanishads.
2. Describe the Gurukul system and evaluate Sanskrit's role in ancient scientific literature and language processing.
3. Explain and apply ancient Indian scientific theories in physics, chemistry, and mathematics
4. Analyze and evaluate ancient wellness systems like Yoga and Ayurveda for their modern relevance.
5. Adentify and illustrate the development of Indian science, technology, engineering, and fine arts.

Unit 1: **Introduction to Indian Knowledge Systems**

Meaning**,** Nature, Scope and Salient Aspects of Bharatiya Jnana Parampara - Introduction to Vedas, Upanishads, Vidya, Kala, Jnana, Shastra - Practices and Continuity of Tradition

Unit 2: Overview of History of Indian Education and Scientific Literature

Gurukul System - Role of Sanskrit in Natural Language Processing - Scientific Literature - Vedic Literature - Available Scientific Treatises - Interlinkings

Unit 3: Introduction to Scientific Theories from Pure Sciences from Ancient Indian Knowledge Systems

Overview of theories from available ancient Indian Literature about Physics, Chemistry and Mathematics - Interlinkings and applications

Unit 4: Introduction to Ancient Indian Wellness Systems

Concept of Wellness – Yoga System - Ayurveda System - Ancient Indian Aesthetics

Unit 5: Development of Engineering, Science, Technology & Fine Arts in India

Various Industries - Silk, Cotton and Ship Building - Evolution of Indian Fine Arts – Cave and Temple Architecture, Vastu - Vidya, Sculpture, Forts and Stepwells, Observatories and Paintings - Music and Natyakala - Cultural Traditions & Folk Arts

❖ **Pedagogy for Teachers: Apart from Class Room Instruction, the following Methods are Suggested.**

1. Project based activities and learning.
2. Presentation and case studies.
3. Film screening and book reviews.

4. Visit to historical places, archives centre, research centre or library nearby.

***Note**: Activities mentioned above are only suggestive. Teacher-educators should encourage students to be innovative.*

Suggested Readings:

1. B. Mahadevan, Bhat Vinayak and Nagendra Pavan R.N., (2022) '*Introduction to Indian Knowledge Systems: Concepts and Applications*' PHI learning PVT, New Delhi ISBN [9789391818203]
2. Dharmapal (1971) '*Indian Science and Technology in the Eighteenth Century*'. Other India Press, Goa.
3. Kapil Kapoor, Singh Avdhesh Kumar, (2005) '*Indian Knowledge Systems*' D.K. Printworld (P) Ltd. ISBN 10: 8124603367 / ISBN 13: 9788124603369
4. Chakradeo, Ujwala, Temples of Bharat, Aayu Publications, New Delhi, 2024.
5. D.N. Bose, S.N. Sen and B. V. Subbarayappa, *A Concise History of Science in India*, Indian National Science Academy, New Delhi, 2009.
6. Datta B. and A. N. Singh, *History of Hindu Mathematics: Parts I and II*, Asia Publishing House, Bombay, 1962.
7. Kapoor, K. (2021), *Indian Knowledge System: Nature, Philosophy, Character in Indian Knowledge System*, vol. 1, Pub. Indian Institute of Advanced Studies, Shimla
8. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Philosophical Systems, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.
9. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Knowledge: Framework and Classification, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.

Video Resources:

1. Introductory lectures by Prof. Gauri Mahulikar.
2. Introductory lectures by Prof. Kapil Kapoor.

Websites:

- https://iksin dia.org/index.php.
- Official Website of IKS- Indian Knowledge System.
- https://www.youtube.com/watch?v=uKcf-hSlcUE.
- Address by Prof Kapil Kapoor | Indian Institute of Advanced Study (FDP 2021).
- https://www.youtube.com/watch?v=MDJTXNiH2_A.
- Mukul Kanitkar on Bharatiya Knowledge System.
- https://www.youtube.com/watch?v=uARMhv97pjk.
- https://www.youtube.com/watch?v=oTwgf56GbsA.
- Scientific History of India | Mukul Kanitkar Lecture in DTU.
- https://youtu.be/gNJNmPJqXJc?si=WFBbuUT65mLZzpOW.
- Ancient India's Scientific Achievements & Contribution in Mathematics, Astronomy, Science & Medicine.

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
OPERATING SYSTEMS CONCEPTS LAB

Course Code: GR25A2090 **L/T/P/C: 0/0/3/1.5**

II Year II Semester

Prerequisite:

- Basics of C Programming / GNU Compiler
- Fundamentals of Data Structures and Algorithms

Course Outcomes:

1. Demonstrate proficiency in Unix/Linux commands and shell scripting for file management, process control, and system automation.
2. Apply C programming to simulate process scheduling algorithms includes FCFS, SJF, Priority, Round Robin and analyze their performance.
3. Implement inter-process communication mechanisms of shared memory, pipes, message queues, and semaphores to coordinate concurrent processes.
4. Simulate and analyze deadlock detection and avoidance algorithms including Deadlock detection, Banker's algorithm for safe resource allocation.
5. Design and evaluate memory management techniques likes First Fit, Best Fit, Page replacement algorithms – FIFO, LRU for efficient resource utilization.

TASK-1: Unix Commands Experiment: Practice Unix commands related to files, directories, data manipulation, and network communication.

Real Example: File navigation and log file monitoring.

Use Case: Daily system administration, server management.

TASK-2: Shell Programming & vi Editor Experiment: Write programs using shell scripts and practice editing with vi.

Real Example: Automating backups and log rotation.

Use Case: System automation and DevOps.

TASK-3: Scheduling Algorithms Experiment: Simulate scheduling algorithms using C:

- FCFS (First Come First Serve)
- SJF (Shortest Job First)
- Priority Scheduling
- Round Robin

Real Example: CPU process scheduling.

Use Case: OS design and process management.

TASK-4: Shared Memory Experiment: Write a C program to implement shared memory between processes.

Real Example: Two processes exchanging data in real-time.

Use Case: Fast communication in multi-process applications.

TASK-5: Threads & Multi-threading Experiment: Simulate thread and multi-thread concepts using C.

Real Example: Multi-threaded web server handling concurrent users.

Use Case: Concurrency and performance optimization.

TASK-6: Inter Process Communication (IPC) Experiment: Write a C program to implement IPC (pipes, message queues).
Real Example: Chat systems between processes.
Use Case: Communication in distributed applications.

TASK-7: Deadlock Detection Experiment: Implement an algorithm for deadlock detection in C.
Real Example: Detecting deadlocks in transaction processing systems.
Use Case: OS resource management.

TASK-8: Banker's Algorithm Experiment: Simulate the Banker's Algorithm for deadlock avoidance in C.
Real Example: Safe state allocation in database systems.
Use Case: Resource allocation in OS and databases.

TASK-9: Readers–Writers Problem Experiment: Simulate Readers-Writers problem using semaphores.
Real Example: Multiple users reading data while only one updates.
Use Case: Synchronization in DBMS.

TASK-10: Memory Management (Fit Algorithms) Experiment: Implement memory allocation strategies in C:

- First Fit
- Best Fit
  Real Example: Dynamic memory allocation in OS.
  Use Case: Efficient memory usage.

TASK-11: Page Replacement Algorithms Experiment: Simulate page replacement algorithms in C:

- FIFO (First In First Out)
- LRU (Least Recently Used)
  Real Example: Virtual memory management.
  Use Case: Improving system performance.

TASK-12: Indexing & Hashing Experiment: Implement indexing and hashing using C.
Real Example: Database indexing for fast search.
Use Case: Information retrieval and database systems.

TEXTBOOKS:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). Operating System Concepts Essentials (2nd ed.). Wiley.

REFERENCE BOOKS:

1. Stallings, W. (2021). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
2. Crowley, C. P. (1997). Operating Systems: A Design-Oriented Approach (1st ed.). McGraw-Hill.
3. Nutt, G. J. (2004). Operating Systems: A Modern Perspective (3rd ed.). Addison-Wesley.
4. Bach, M. J. (1986). The Design of the UNIX Operating System. Prentice-Hall.
5. Bovet, D. P., & Cesati, M. (2005). Understanding the Linux Kernel (3rd ed.). O’Reilly Media

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
PRINCIPLES OF SOFTWARE ENGINEERING LAB

Course Code: GR25A2091 **L/T/P/C: 0/0/3/1.5**

II Year II Semester

Pre-requisite: Basic knowledge of programming for problem solving.

Course Outcomes:

1. Demonstrate the ability to analyze real-time problems and extract precise software requirements.
2. Develop and implement appropriate software design models to address system requirements.
3. Apply modern engineering tools effectively for system specification, modeling, design, and implementation.
4. Devise suitable software engineering methodologies to deliver solutions for real-time applications.
5. Construct and evaluate test cases to validate solutions for real-time problems.

Task 1: Prepare the problem statement

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- The Unified Library System reduces manual effort, ensures accuracy, and enhances efficiency and user satisfaction through a centralized automated solution.
- The Online Railway Reservation System improves accessibility and provides a fast, accurate, and user-friendly ticketing platform while eliminating manual inefficiencies.

Use Case:

- Unified Library System Use Case : Manage user registration, book search, issue/return, renewals, reservations, fines, and catalog maintenance.
- Online Railway Reservation System Use Case : Handle user registration, train search, booking, cancellation, payment, ticket generation, and schedule management.

Task 2: Develop Software Requirement Specification (SRS)

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies, Specific Requirements, System Models

Use Case: Use Cases in SRS show “what the system should do” from the perspective of users

Task 3: Design the data flow diagram

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- Unified Library System
  - User Management
  - Catalog Management
  - Borrow/Return Book
  - Renewal/Reservation
  - Fine Calculation
  - Reports Generation
- Online Railway Reservation System.
  - User Registration & Login
  - Search & Availability
  - Book Ticket
  - Cancel Ticket
  - Delivery Route Optimization

Use Case: DFD can be used to represent the main process(es), data stores, entities and their relationships of any system.

Task 4: Design the class diagrams

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- The objective of a Class Diagram is to provide a clear and structured representation of the system's classes and their relationships, serving as the foundation for object-oriented design and implementation.

Use Case:

- Class Diagram can be used to model the static view of a system by showing classes, their properties, operations, and relationships, which guides system implementation and communication among stakeholders.

Task 5: Design the Use-case diagram

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- The **objective of a Use Case Diagram** is to capture the functional requirements of a system by representing the interactions between external actors and the system, helping to understand what the system should do from a user's perspective.

Use Case:

- Use case Diagram can be used to model the dynamic view of a system by representing the actors, use cases and collaborations of any system.

Task 6: Design the interaction diagrams

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- ATM system and user interactions

Use Case:

- **Interaction Diagram** is to describe the flow of messages between objects and actors, helping developers understand the sequence of operations and the communication structure required for implementing system behavior.

Task 7: Perform Forward Engineering and generate a report

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- Banking, E-commerce, ATM, Library, Hospital systems.

Use Case:

- Convert the model (class diagram) into code and database structures for faster, consistent, and error-free system implementation.

Task 8: Perform Reverse Engineering and generate a report

Experiment:

- Unified Library System
- Online Railway Reservation System.

Real Example:

- System modernization, maintenance, migration, documentation, security analysis.

Use Case:

- Convert existing code or system artifacts into a class diagram to understand design, support maintenance, or guide system for enhancement.

TASK 9: Program on control statements

Experiment: Demonstrate the working process of control statements.

Real Example:

- Banking, Reservations, Libraries, E-commerce.

Use Case: Decision-making and repetitive tasks in real-world applications can be implemented with relevance to systems such as ATM systems, library management, online reservations.

Task 10: Create a test plan document

Experiment: Unified Library System

Real Example:

- Bug identification in ATM transaction flows.

Use Case:

- Each test case corresponds to a use case that defines the functionality being tested, justifies test coverage of system requirements, and serves as essential documentation for labs and practical exams.

Task 11: Implement a Junit Test program and design test cases
Experiment: Program to find the maximum of an arrayof numbers.
Real Example:

- Finding the maximum of an array is commonly used in real-world systems such as banking (highest transaction), e-commerce (costliest product), healthcare (peak vital signs), education (top exam score), and sports (highest player performance).

Use Case: Array of numeric values (stock prices, temperatures, marks, scores, readings, or prices

Task 12: Implement a Junit Test programand design test cases
Experiment: Program to count the number of elements in array of numbers.
Real Example:

- Counting the number of elements in an array is widely applied in real-world systems such as banking (number of transactions), e-commerce (items in a cart), healthcare (daily readings recorded), education (questions answered or subjects taken), and social media (likes, comments, or posts made).

Use Case: Counting the number of elements in an array helps in understanding how systems track data such as total marks obtained in exams, number of subjects enrolled, assignments submitted, books borrowed from the library, or even the number of files/projects saved on their computer**.**

TEXTBOOKS:

1. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson,
2. Pearson Education.
3. UML 2 Toolkit, Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, WILEYDreamtech India Pvt. Ltd.
4. Software Engineering, Ian Sommerville

REFERENCE BOOKS:

1. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino
2. Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices, Michael Jackson
3. Software Metrics: A Rigorous and Practical Approach, Norman E Fenton, Shari Lawrence Pfleeger
4. Software Engineering: Theoryand Practice, Shari Lawrence Pfleeger and Joanne M. Atlee
5. Object-Oriented Software Construction, Bertrand Meyer

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ALGORITHM DESIGN AND ANALYSIS LAB

Course Code: GR25A2092 **L/T/P/C: 0/ 0/ 3/1.5**

II Year II Semester

Prerequisites: Data Structures and Algorithms, Basic concepts of C or Java

Course Outcomes:

1. Demonstrate proficiency in evaluating algorithms using asymptotic notations, including best, average, and worst case time/space complexities, and solving related recurrence relations.
2. Design and Execute various algorithmic strategies divide and conquer, identifying suitable use cases and demonstrating their application.
3. Apply various algorithmic strategies dynamic programming and greedy approach, identifying suitable use cases and demonstrating their application.
4. Apply and Evaluate various algorithmic strategies backtracking and branch & bound, identifying suitable use cases and demonstrating their application.
5. Compare and Classify tractable (polynomial-time) and intractable (super-polynomial or exponential-time) problems; identify and classify problems as P, NP, NP-hard, or NP-complete, and assess their relationships through polynomial-time reductions and Cook's theorem

Lab Experiments

TASK 1

Binary Search & Quick Sort

Experiment: Implement and analyse time complexity in best & worst case for Binary Search and Quick Sort.

Real Example:

Binary Search: Searching for a product in a sorted e-commerce inventory. Quick Sort: Sorting transaction amounts or product prices for recommendation systems.

Use Case: Used in search engines, online shopping platforms, and large-scale data sorting.

TASK 2

Merge sort & Strassen Matrix Multiplication

Experiment: Implement and analyse time complexity in best & worst case for Merge sort and Strassen Matrix Multiplication.

Real Example:

Merge sort: Sorting student grades for merit lists.

Strassen Multiplication: Fast matrix multiplication for image processing and scientific computations.

Use Case: Essential in data analytics, graphics, and high-performance computing.

TASK 3

Greedy Algorithms

Experiment: Implement and analyse time complexity of Greedy Application Problems.

Real Example:

Coin Change Problem: Minimizing the number of coins in an ATM.
Activity Selection: Scheduling maximum tasks in limited time.
Use Case: Used in financial systems, resource allocation, and scheduling.

TASK 4
Dynamic Programming
Experiment: Implement and analyze time complexity of Dynamic Programming Application Problems.
Real Example:
Shortest Path (Floyd-Warshall): GPS route optimization.
Knapsack Problem: Optimal selection of goods underweight constraints.
Use Case: Applied in logistics, supply chain optimization, and bioinformatics.

TASK 5
Greedy And Minimum Spanning Tree (Prim's & Kruskal's)
Experiment: Implement and analyse time complexity of Greedy Application Problems, Prim's & Kruskal's Algorithms.
Real Example:
Network Design: Laying minimum cost network cables across cities.
Use Case: Telecom networks, transportation planning, and infrastructure design.

TASK 6
Backtracking
Experiment: Implement and analyse time complexity of Backtracking Application Problems.
Real Example:
N-Queens Problem: Placing queens on a chessboard without attacking each other.
Sudoku Solver: Filling puzzle grids systematically.
Use Case: Game development, puzzle solving, AI-based problem-solving.

TASK 7
Branch & Bound
Experiment: Implement and analyse time complexity of Branch & Bound Application Problems.
Real Example:
Travelling Salesman Problem: Finding shortest route for delivery services.
Use Case: Logistics, route optimization, and supply chain management.

TASK 8
BFS & DFS
Experiment: Implement and analyze time complexity of BFS and DFS and their applications.
Real Example:
Campus Map Navigator: BFS for shortest path, DFS for exploring all buildings.
Use Case: Social media analysis, route planning, graph-based search.

TASK 9
Dijkstra & Floyd-Wars hall Algorithms
Experiment: Implement and analyse time complexity of Dijkstra and Floyd Warshall

Algorithms.

Real Example:

Navigation Systems: Dijkstra for shortest path, Floyd-Warshall for all-pairs shortest paths.

Use Case: GPS systems, traffic routing, and network packet routing.

TASK 10

Topological Sorting And Network Flow

Experiment: Implement and analyse time complexity of Topological Sorting, Network Flow Problems.

Real Example:

Topological Sort: Task scheduling with dependencies (e.g., course prerequisites).

Network Flow: Maximum flow in data networks or pipelines.

Use Case: Project management, compiler optimization, and data flow systems.

TASK 11

P, NP, NP-Complete, NP-Hard

Experiment: Implement sample problem on P, NP, NP-complete, and NP-hard.

Real Example:

Subset Sum Problem: Checking if a subset with a given sum exists.

Use Case: Cryptography, optimization problems, computational complexity research.

TASK 12

Randomized Quick Sort

Experiment: Implement and analyse time complexity of Randomized Quick Sort.

Real Example:

Random Shuffling: Sorting large data sets with randomized pivots for performance optimization.

Use Case: Used in large-scale sorting where uniform distribution of pivot reduces worst-case performance.

TEXTBOOKS:

1. Introduction to Algorithms, 3rd Edition, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT Press, 2009.
2. Algorithm Design, by Jon Kleinberg and Éva Tardos, Pearson, 2006.
3. Fundamentals of Computer Algorithms, by Ellis Horowitz, Sartaj Sahni, and S. Rajasekaran, Universities Press, 2007.

REFERENCE BOOKS:

1. Design and Analysis of Algorithms, by Anany Levitin, Pearson Education, 3rd Edition, 2012.
2. Algorithms, by Robert Sedgewick and Kevin Wayne, Addison-Wesley, 4th Edition, 2011.
3. The Design and Analysis of Algorithms, by Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, Pearson, 1974.
4. Algorithmic Puzzles, by Anany Levitin and Maria Levitin, Oxford University Press, 2011.
5. Data Structures and Algorithm Analysis in C, by Mark Allen Weiss, Pearson, 2nd Edition, 1996.
6. Approximation Algorithms, by Vijay V. Vazirani, Springer, 2003.