**Scheme of Teaching and Examinations-2025**
Outcome-Based Education (OBE) and Choice-Based Credit System (CBCS)
(Effective from the academic year 2025-26)

**I Semester** **(Chemistry Group)**

| Sl. No | Course and Course Code | | Course Title | TD/PSB | Teaching Hours/Week | | | | Examination | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Theory Lecture | Tutorial | Practical/ Drawing | SAAE | Duration in hours | CIE Marks | SEE Marks | Total Marks | |
| | | | | | L | T | P | S | | | | | |
| 1 | ASC | 1BMATx101 | Applied Mathematics-I (Stream Specific) | Maths Dept | 3 | 2 | 0 | | 03 | 50 | 50 | 100 | 04 |
| 2 | ASC(IC) | 1BCHEx102 | Applied Chemistry (Stream Specific) | CHE Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 3 | ETC | 1BAIA103/ BETC105x | Introduction to AI and Applications | Any Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 4 | ESC | 1BESC104x | Engineering Science Course I | Respective Engg Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 5 | PLC(IC) | 1BPLC105x | Programming Language Course | CSE & allied Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 6 | AEC | 1BENG106 | Communication Skills | Humanities Dept | 1 | 0 | 0 | | 02 | 50 | 50 | 100 | 01 |
| 7 | AEC (NCMC) | 1BICO107 | Indian Constitution & Engineering Ethics | Humanities Dept | 1 | 0 | 0 | | -- | 100 | -- | 100 | PP |
| 8 | AEC/SDC | 1BIDTL158 | Innovation and Design Thinking Lab **(Project-based learning)** | Any Dept | 0 | 0 | 2 | | 02 | 50 | 50 | 100 | 01 |
| | **TOTAL** | | | | **17** | **02** | **07** | | **20** | **450** | **350** | **800** | **20** |
| **9** | **AICTE Activity Points** (students have to earn **100** activity points between 01 to 08 semesters) | | | | Compulsory requirement for the award of a degree | | | | | | | | |

**ASC**-Applied Science Course, **ESC**- Engineering Science Courses, **IC** – Integrated Course (Practical Course Integrated with Theory Course), **PLC(IC)**- Programming Language Course (Integrated Course), **AEC**- Ability Enhancement Course, **NCMC:** Non Credit Mandatory Course, **AEC/SDC**- Ability Enhancement Course/Skill Development course, **TD/PSB**- Teaching Department / Paper Setting Board, **S- (SAAE)-**Students' Academic Activity Engagement Hours, **CIE** –Continuous Internal Evaluation, **SEE**- Semester End Examination, **PP** : (Pass/Pass) is assigned to a noncredit course. "PP" represents pass in a course provided students have successfully completed the CIE requirements. Otherwise, "NP-not pass shall be awarded. "PP" is essential for the award of the degree

| **Credit Definition:** | |
|---|---|
| 1-hour Lecture **(L)** per week=**1Credit** | 04-Credit courses are designed for 50 hours of Teaching-Learning sessions |
| 2-hoursTutorial**(T)** per week=**1Credit** | 04-Credit (IC) courses are designed for 40 hours' theory and 10-12 hours of practical sessions |
| 2-hours Practical / Drawing (**P**) per week=**1Credit** | 03-Credit courses are designed for 40 hours of Teaching-Learning Session |
| | 02- Credit courses are designed for 25 hours of Teaching-Learning Session |
| | 01-Credit courses are designed for 12 hours of Teaching-Learning sessions |

| Applied Mathematics-I | | | | | Applied Chemistry | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Title** | **L** | **T** | **P** | **Code** | **Title** | **L** | **T** | **P** |
| 1BMATC101 | Differential Calculus and Linear Algebra: CV Stream | 3 | 2 | 0 | 1BCHEC102 | Applied Chemistry for Sustainable Structure & Material Design (CV) | 3 | 0 | 2 |
| 1BMATM101 | Differential Calculus and Linear Algebra: ME Stream | 3 | 2 | 0 | 1BCHEM102 | **Applied Chemistry for Advanced Metal Protection and Sustainable Energy Systems** (ME) | 3 | 0 | 2 |
| 1BMATE101 | Differential Calculus and Linear Algebra; EEE stream | 3 | 2 | 0 | 1BCHEE102 | **Applied Chemistry for Emerging Electronics and Futuristic Devices** (EEE, ECE) | 3 | 0 | 2 |
| 1BMATS101 | Calculus And Linear Algebra: CSE stream | 3 | 2 | 0 | 1BCHES102 | Applied Chemistry for Smart Systems (CSE) | 3 | 0 | 2 |
| | | | | | | | | | |
| **Engineering Science Courses-I (ESC-I)** | | | | | **Programming Language Courses (PLC)** | | | | |
| **Code** | **Title** | **L** | **T** | **P** | **Code** | **Title** | **L** | **T** | **P** |
| 1BESC104A | Building Sciences & Mechanics | 3 | 0 | 0 | 1BPLC105E | Introduction to C Programming (For none IT programmes) | 3 | 0 | 0 |
| 1BESC104B | Introduction to Electrical Engineering | 3 | 0 | 0 | 1BPLC105B | Python Programming (for CSE and allied programmes) | 3 | 0 | 0 |
| 1BESC104C | Introduction to Electronics and Communication Engineering | 3 | 0 | 0 | | | | | |
| 1BESC104D | Introduction to Mechanical Engineering | 3 | 0 | 0 | | | | | |
| 1BESC104E | Essentials of Information Technology | 3 | 0 | 0 | | | | | |

**Integrated courses (IC), combining theory with practical components.**

(i) Theory sessions shall be conducted for 3 hours per week, while the practical sessions shall be conducted for 2 hours per week.

(ii)Theory components shall be evaluated through both Continuous Internal Evaluation (CIE) and Semester End Examination (SEE).

(iii)The practical component shall be assessed only through CIE.

**The Mathematics/Chemistry courses shall be taught by a single faculty member per session, with no sharing of the course (subject) modules.** The **tutorial sessions** for the **mathematics course** shall be conducted in the laboratory environment using **Maxima/Mathematica/ Python/Scilab/MATLAB software** to enhance computational understanding and application skills.

All students admitted to the engineering program have to complete **Applied Mathematics-I and Applied Mathematics-II** in I and II semesters by selecting the subjects prescribed for their stream, viz. CV, ME, EEE or CSE, under the heading Mathematics-I and Mathematics-II.

Those who have completed the chemistry course under the heading Applied Chemistry in I semester have to select the prescribed stream wise physics course under the heading Applied physics during II semester.

**Engineering Sciences Courses-I (ESC-I):** These courses are designed to broaden the technical knowledge of students beyond their core area of study. These courses enable students to gain a foundational understanding of engineering principles from other stream courses. Students are required to select and complete two courses that are not belong to their admitted program stream. For example, a student admitted to the any programme of the Civil Engineering stream should not select Introduction to Building Sciences but any other two. One course shall selected under ESC-I and another course under ESC-II. The two courses must be different from the other.

**Communication Skills:** This course shall be conducted in a laboratory environment

The **Student Induction Programme** (SIP), initiated by the All India Council for Technical Education (AICTE), is designed to help newly admitted students in technical institutions transition smoothly into the higher education environment. It aims to familiarize students with the institutional culture, foster connections with peers and faculty, and provide a foundation for holistic learning. Activities under SIP may include Physical Activities, Creative Arts, Universal Human Values, Literary Events, and Proficiency Modules. Lectures shall be by Eminent Personalities, Local Area Visits, Department/Branch Familiarization, and Innovation-related sessions.

The first year of the Engineering programmes is composed of I semester, II semester and Summer Semester. SIP activities shall be scheduled in the afternoon sessions during the first week of class commencement of I and II semesters only.

The specific programmes to be conducted will be notified separately by the University via the academic calendar or through a separate notification.

**AICTE Activity Points Requirement for BE/B.Tech. Programmes**

As per AICTE guidelines (refer Chapter 6 – *AICTE Activity Point Program, Model Internship Guidelines*), in addition to academic requirements, students must earn a specified number of **Activity Points** to be eligible for the award of the degree. The points to be earned is:

1. **Regular students** admitted to a 4-year degree program must earn **100 Activity Points**.
2. **Lateral entry students** (joining from the second year) must earn **75 Activity Points**.
3. **Students transferred** from other universities directly into the fifth semester must earn **50 Activity Points** from the date of entry into VTU.

These Activity Points are **non-credit** and will not be considered for **the SGPA/CGPA** or be used for **vertical progression**. However, earning Activity Points is mandatory for the **award of the degree**, and the points earned will be reflected on the **eighth semester Grade Card**.

If a student completes all the semesters (eight or six) at the end of the programme but fails to earn the required Activity Points, the eighth-semester Grade Card will be withheld until the requirement is fulfilled. Also, the degree will be awarded only after the Grade Card has been released.

The hours spent earning the activity points will not be counted for regular attendance requirements. Students can accumulate these points at any time during their program period, including weekends, holidays, and vacations, starting from the year of admission, provided they meet the minimum hours of engagement prescribed for each activity by AICTE.

| Sl. No | Stream | UG Programmes under the stream with code |
|---|---|---|
| 1 | Civil Engineering Stream (CV) | (1) Civil engineering (CV), (2) Mining Engineering (MI) |
| 2 | Mechanical Engineering Stream ME | (1)Aeronautical Engineering (AE), (2)Aerospace Engineering (AS),(3) Agreecultural Engineering (AG),(4)Automation and Robotics (AR), (5)Automobile Engineering (AU), (6)Chemical Engineering (CH), (7) Industrial & Production Engineering (IP), (8)Industrial Engineering & Management (IM), (9) Manufacturing Science and Engineering (MS), (10) Marine Engineering (MR), (11) Mechanical & Smart Manufacturing (MM), (12) Mechanical Engineering (ME), (13)Mechatronics (MT), (14) Petrochem Engineering (PC), (15)Robotics & Automation (RA),(16) Robotics and Artificial Intelligence (RI),(17)Silk Technology (ST), (18) Textile Technology (TX),(19)Energy Engineering (ER),(20) Smart Agritech (SA). |
| 3 | Electrical and Electronics Engineering Stream (EEE) | (1)Electronics & Communication Engineering (EC), (2)Biomedical Engineering (BM), (3)Electrical & Electronics Engineering (EE), (4) Electronics & Instrumentation Engineering (EI),(5) Electronics & Telecommunication Engineering (ET),(6) Industrial IoT (IO), (7) Medical Electronics Engineering (ML),(8) Electronics Engineering (VLSI Design and Technology) (VL),(9) Electronics & Communication( Advanced Communication Technology) (EA),(10) Electronics & Computer Engineering (UE). |
| 4 | Computer Science and Engineering Stream (CSE) | (1) Computer Science and Engineering (CS), (2)Computer Engineering (CE), (3) Artificial Intelligence and Data Science (AD), (4)Artificial Intelligence and Machine Learning (AI),(5)Biotechnology (BT),(6)Computer & Communication Engineering (CM), (7) Computer Science and Business System (CB),(8)Computer Science and Design (CG),(9)Computer Science and Engineering (IoT) (CO), (10)CSE(Artificial Intelligence and Machine Learning) (CI),(11) CSE(Artificial Intelligence) (CA),(12) CSE(Cyber Security) (CY), (13)CSE(Data Science) (CD),(14) CSE(IoT and Cyber Security including Block Chain Technology) (IC), (15) Data Science (DS), (16) Information Science & Engineering (IS),(17) Computer Science (CR). |
| | | |

**Scheme of Teaching and Examinations-2025**
Outcome-Based Education(OBE) and Choice Based Credit System(CBCS)
(Effective from the academic year 2025-26)

**II Semester (For the students who have studied the Chemistry group in I semester)**

| Sl. No | Course and Course Code | | Course Title | TD/PSB | Teaching Hours/Week | | | | Examination | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Theory Lecture | Tutorial | Practical/ Drawing | SAAE | Duration in hours | CIE Marks | SEE Marks | Total Marks | |
| | | | | | L | T | P | S | | | | | |
| 1 | ASC | 1BMATx201 | Applied Mathematics -II (Stream Specific) | Maths Dept | 3 | 2 | 0 | | 03 | 50 | 50 | 100 | 04 |
| 2 | ASC(IC) | 1BPHYx202 | Applied Physics (Stream Specific) | PHY Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 3 | ESC | 1BCEDx203 | Computer-Aided Engineering Drawing (Stream Specific) | ME dept | 2 | 0 | 2 | | 03 | 50 | 50 | 100 | 03 |
| 4 | ESC | 1Bxxx204x | Engineering Science Course-II | Respective Engg Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 5 | PSC | 1Bxxx205 | Programme Specific Courses | Respective Engg Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 6 | AEC (NCMC) | 1BSKS206 | Soft Skills | Humanities Dept | 1 | 0 | 0 | | -- | 100 | --- | 100 | PP |
| 7 | PSC/ESC | 1Bxxxl207x | Program-Specific Course Lab | Respective dept | 0 | 0 | 2 | | 02 | 50 | 50 | 100 | 01 |
| 8 | AEC/SDC | 1BPRJ258 | Interdisciplinary Project-Based Learning | Combination of Departments | 0 | 0 | 0 | 02 | 02 | 50 | 50 | 100 | 01 |
| 9 | HSMC | 1BKSK209(BKSK107)/ 1BKBK209(BKBK107) | Samskrutika Kannada/ Balake Kannada | Humanities Dept | 1 | 0 | 0 | | 01 | 50 | 50 | 100 | 01 |
| | | | | **TOTAL** | **16** | **02** | **06** | | **21** | **500** | **400** | **900** | **20** |

**ASC**-Applied Science Course, **IC** – Integrated Course (Practical Course Integrated with Theory Course), **ESC**- Engineering Science Courses, **PSC**-Programme Specific Course, **ESC**- Engineering Science Courses, **ETC**- Emerging Technology Course, **AEC**- Ability Enhancement Course, **NCMC:** Non Credit Mandatory Course, **PP** : (Pass/Pass) is assigned to a non credit course. "PP" represents pass in course provided students have successfully completed the CIE requirements. Otherwise, "NP-not pass shall be awarded. "PP" is essential for the award of the degree **HSMC**-Humanity, Social Science and management Course, **AEC/SDC**- Ability Enhancement Course/Skill Development course, **TD/PSB**- Teaching Department / Paper Setting Board, **CIE** –Continuous Internal Evaluation, **SEE**- Semester End Examination, **S- (SAAE)-**Students' Academic Activity Engagement Hours,

| Applied Mathematics-II | | | | | Applied Physics | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Title** | **L** | **T** | **P** | **Code** | **Title** | **L** | **T** | **P** |
| 1BMATC201 | Differential Calculus and Numerical Methods: CV stream | 3 | 2 | 0 | 1BPHYC202 | Physics for Sustainable Structural Systems (CV stream) | 3 | 0 | 2 |
| 1BMATM201 | Multivariable Calculus and Numerical Methods: ME stream | 3 | 2 | 0 | 1BPHYM202 | Physics of Materials (Mech stream) | 3 | 0 | 2 |
| 1BMATE201 | Calculus, Laplace Transform, and Numerical Techniques: EEE stream | 3 | 2 | 0 | 1BPHEC202 | Quantum Physics and Electronic Sensors (ECE stream) | 3 | 0 | 2 |
| 1BMATS201 | Numerical Methods: CSE Stream | 3 | 2 | 0 | 1BPHEE202 | Electrical Engineering Materials (EEE) | 3 | 0 | 2 |
| | | | | | 1BPHYS202 | Quantum Physics and Applications (CSE stream) | 3 | 0 | 2 |
| **Programme Specific Courses (PSC)** | | | | | **Programme Specific Courses Lab (PSCL)** | | | | |
| 1BCIV205 | Engineering Mechanics | 3 | 0 | 0 | 1BMEML207 | Mechanics and Materials Lab | 0 | 0 | 2 |
| 1BEME205 | Elements of Mechanical Engineering | 3 | 0 | 0 | 1BEMEL207 | Elements of Mechanical Engineering Lab | 0 | 0 | 2 |
| 1BBEE205 | Basics of Electrical Engineering | 3 | 0 | 0 | 1BBEEL207 | Basic Electrical Lab | 0 | 0 | 2 |
| 1BECE205 | Fundamentals of Electronics & Communication Engineering | 3 | 0 | 0 | 1BECEL207 | Fundamentals of Electronics & Communication Engineering Lab | 0 | 0 | 2 |
| 1BEIT205 | Programming in C | 3 | 0 | 0 | 1BPOPL207 | C Programming Lab | 0 | 0 | 2 |
| 1BEBT205 | Elements of Biotechnology and Biomimetics | 3 | 0 | 0 | 1BSSAL207 | Soil Science and Agronomy Field Lab | 0 | 0 | 2 |
| 1BSSA205 | Principles of Soil Science and Agronomy | 3 | 0 | 0 | 1BEBTL207 | Elements of Biotechnology Lab | 0 | 0 | 2 |
| 1BEAE205 | Elements of Aeronautical Engineering | 3 | 0 | 0 | 1BEAEL207 | Elements of Aeronautical Engineering Lab | 0 | 0 | 2 |
| 1BECHE205 | Elements of Chemical Engineering | 3 | 0 | 0 | 1BECHEL207 | Elements of Chemical Engineering Lab | 0 | 0 | 2 |
| **Engineering Science Courses-II (ESC-II)** | | | | | **Computer-Aided Engineering Drawing** | | | | |
| 1BESC204A | Building Sciences & Mechanics | 3 | 0 | 0 | **Code** | **Title** | **L** | **T** | **P** |
| 1BESC204B | Introduction to Electrical Engineering | 3 | 0 | 0 | 1BCEDC203 | Computer-Aided Engineering Drawing for CV Stream | 2 | 0 | 2 |
| 1BESC204C | Introduction to Electronics & Communication Engineering | 3 | 0 | 0 | 1BCEDM203 | Computer-Aided Engineering Drawing for ME stream Engineering | 2 | 0 | 2 |
| 1BESC204D | Introduction to Mechanical Engineering | 3 | 0 | 0 | 1BCEDEC203 | Computer-Aided Engineering Drawing for EEE stream | 2 | 0 | 2 |
| 1BESC204E | Essentials of Information Technology | 3 | 0 | 0 | 1BCEDEE203 | Computer-Aided Engineering Drawing for EEE stream (Only for EEE Students) | 2 | 0 | 2 |
| | | | | | 1BCEDS203 | Computer-Aided Engineering Drawing for CSE stream | 2 | 0 | 2 |

**Integrated courses (IC), combining theory with practical components.**

(i)Theory sessions will be conducted for 3 hours per week, while the practical sessions will be conducted for 2 hours per week.

(ii)Theory component shall be evaluated through both Continuous Internal Evaluation (CIE) and Semester End Examination (SEE).

(iii)The practical component will be assessed only through CIE.

**The Mathematics/Physics courses shall be taught by a single faculty member per session, with no sharing of the course (subject) modules.** The **tutorial sessions** for the **mathematics course** shall be conducted in a laboratory environment using **Maxima/Mathematica/ Python/Scilab/MATLAB software** to enhance

| |
|---|
| computational understanding and application skills. |
| Students admitted to a specific engineering stream are required to select and successfully complete **Applied Mathematics-I** and **Applied Physics courses** that are aligned to their program stream. |

| |
|---|
| **Programme Specific Courses (PSC):** Programme Specific Courses (PSC) are a set of core courses tailored to a specific branch or discipline of engineering in which a student is enrolled (e.g., Mechanical Engineering, Computer Science, Civil Engineering, etc.). These courses are intended to provide students with in-depth knowledge and specialized skills essential for professional competence in the chosen field.<br>Students must select and complete the course from this group that **corresponds to their admitted program stream**.<br>Similarly, students are also required to choose and pass laboratory courses that are specific to their stream from the **Programme Specific Courses Laboratory (PSCL)** group. |
| **Computer-Aided Engineering Drawing:** The courses under this category are stream-specific. Students must select and complete the course that corresponds to their admitted engineering stream**.** |
| **Engineering Sciences Courses-II (ESC-II):** These courses are designed to broaden the technical knowledge of students beyond their core area of study. These courses enable students to gain a foundational understanding of engineering principles from other disciplines. Students are required to select and complete a course that does not belong to their admitted program stream. Students should select a course other than that was selected under ESC-I and other than course not belonging to their stream. |
| **F**or the course ***Interdisciplinary Project*** **(BPRJ259)**, it is mandatory to form a team comprising students from multiple engineering disciplines. For example, a project team may include students from Mechanical Engineering, Electronics and Communication Engineering (ECE), and Computer Science and Engineering (CSE), working collaboratively to design and implement the project. |

**Scheme of Teaching and Examinations (2025)**
Outcome-Based Education (OBE)and Choice-Based Credit System (CBCS)
(Effective from the academic year 2025-26)

**I Semester** **(Physic Group)**

| Sl. No | Course and Course Code | | Course Title | TD/PSB | Teaching Hours/Week | | | | Examination | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Theory Lecture | Tutorial | Practical/ Drawing | SAAE | Duration in hours | CIE Marks | SEE Marks | Total Marks | |
| | | | | | L | T | P | S | | | | | |
| 1 | ASC | 1BMATx101 | Applied Mathematics -I (Stream Specific) | Maths Dept | 3 | 2 | 0 | | 03 | 50 | 50 | 100 | 04 |
| 2 | ASC(IC) | 1BPHYx102 | Applied Physics (Stream Specific) | PHY Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 3 | ESC | 1BCEDx103 | Computer-Aided Engineering Drawing (Stream Specific) | ME Dept | 2 | 0 | 2 | | 03 | 50 | 50 | 100 | 03 |
| 4 | ESC | 1BXXX104x | Engineering Science Courses-I | Respective Engg Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 5 | PSC | 1Bxxx105 | Programme Specific Course | Respective Engg dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 6 | AEC (NCMC) | 1BSKS106 | Soft Skills | Humanities Dept | 1 | 0 | 0 | | - | 100 | --- | 100 | PP |
| 7 | PSC | 1BxxxL107 | Program-Specific Course Lab | Respective Engg Dept | 0 | 0 | 2 | | 02 | 50 | 50 | 100 | 01 |
| 8 | AEC/SDC | 1BIDTL158 | Innovation and Design Thinking Lab **(Project-based learning)** | Respective Dept | 0 | 0 | 2 | | 02 | 50 | 50 | 100 | 01 |
| 9 | HSMS | 1BKSK109(BKSK107)/ 1BKBK109(BKBK107) | Samskrutika Kannada/ Balake Kannada | Humanities Dept | 1 | 0 | 0 | | 01 | 50 | 50 | 100 | 01 |
| | | | | **TOTAL** | 16 | 02 | 08 | | 20 | 500 | 400 | **900** | **20** |
| 10 | **AICTE Activity Points** (students have to earn **100** activity points between 01 to 08 semester) | | | | Compulsory requirement for the award of a degree | | | | | | | | |

**ASC**-Applied Science Course, **IC** – Integrated Course (Practical Course Integrated with Theory Course), PSC-Programme Specific Course, **ESC**- Engineering Science Courses, **ETC**-Emerging Technology Course, **AEC**- Ability Enhancement Course, **NCMC:** Non Credit Mandatory Course, **PP** : (Pass/Pass) is assigned to a noncredit course. "PP" represents pass in course provided students have successfully completed the CIE requirements. Otherwise, "NP-not pass shall be awarded. "PP" is essential for the award of the degree. **PLC(IC)**- Programming Language Course (Integrated Course), **AEC/SDC**- Ability Enhancement Course/Skill Development course, **TD/PSB**- Teaching Department / Paper Setting Board, **HSMS**-Humanity, Social Science and management Course, **S- (SAAE)** Students' Academic Activity Engagement Hours, **CIE –**Continuous Internal Evaluation, **SEE-**Semester End Examination,

**Credit Definition:**
1-hour Lecture **(L)** per week=**1Credit**
2-hoursTutorial**(T)** per week=**1Credit**
2-hours Practical / Drawing (**P**) per week=**1Credit**

04-Credit courses are designed for 50 hours of Teaching-Learning Session
04-Credit (IC) is designed for 40 hours' theory and 10-12 hours of practical sessions
03-Credit courses are designed for 40 hours of Teaching-Learning Session
02- Credit courses are designed for 25 hours of Teaching-Learning Session
01-Credit courses are to be designed for 12 hours of Teaching-Learning sessions

| Applied Mathematics-I | | | | | Applied Physics | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Title** | **L** | **T** | **P** | **Code** | **Title** | **L** | **T** | **P** |
| 1BMATC101 | Differential Calculus and Linear Algebra: CV Stream | 3 | 2 | 0 | 1BPHYC102 | Physics for Sustainable Structural Systems (CV stream) | 3 | 0 | 2 |
| 1BMATM101 | Differential Calculus and Linear Algebra: ME Stream | 3 | 2 | 0 | 1BPHYM102 | Physics of Materials (Mech stream) | 3 | 0 | 2 |
| 1BMATE101 | Differential Calculus and Linear Algebra: EEE stream | 3 | 2 | 0 | 1BPHEC102 | Quantum Physics and Electronics Sensors (EEE stream) | 3 | 0 | 2 |
| 1BMATS101 | Calculus and Linear Algebra: CSE Stream | 3 | 2 | 0 | 1BPHEE102 | Electrical Engineering Materials (EEE stream-only for EEE students) | 3 | 0 | 2 |
| | | | | | 1BPHYS102 | Quantum Physics and Applications (CSE stream) | 3 | 0 | 2 |
| **Computer-Aided Engineering Drawing** | | | | | **Engineering Science Courses-I(ESC-I)** | | | | |
| 1BCEDC103 | Computer-Aided Engineering Drawing for CV Stream | 2 | 0 | 2 | 1BESC104A | Building Sciences and Mechanics | 3 | 0 | 0 |
| 1BCEDM103 | Computer-Aided Engineering Drawing for ME stream | 2 | 0 | 2 | 1BESC104B | Introduction to Electrical Engineering | 3 | 0 | 0 |
| 1BCEADEC103 | Computer-Aided Engineering Drawing for EEE stream | 2 | 0 | 2 | 1BESC104C | Introduction to Electronics & Communication Engineering | 3 | 0 | 0 |
| 1BCEADEE103 | Computer-Aided Engineering Drawing for EEE stream(only for EEE students) | 2 | 0 | 2 | 1BESC104D | Introduction to Mechanical Engineering | 3 | 0 | 0 |
| 1BCEDS103 | Computer-Aided Engineering Drawing for CSE stream | | | | 1BESC104E | Essentials of Information Technology | 3 | 0 | 0 |
| **Programme Specific Courses (PSC)** | | | | | **Program-Specific Course Lab (PSCL)** | | | | |
| 1BCIV105 | Engineering Mechanics | 3 | 0 | 0 | 1BMEML107 | Mechanics and Materials Lab | 0 | 0 | 2 |
| 1BBEE105 | Basics of Electrical Engineering | 3 | 0 | 0 | 1BBEEL107 | Basic Electrical Lab | 0 | 0 | 2 |
| 1BECE105 | Fundamentals of Electronics & Communication Engineering | 3 | 0 | 0 | 1BECEL107 | Fundamentals of Electronics & Communication Engineering Lab | 0 | 0 | 2 |
| 1BEME105 | Elements of Mechanical Engineering | 3 | 0 | 0 | 1BEMEL107 | Elements of Mechanical Engineering Lab | 0 | 0 | 2 |
| 1BEIT105 | Programming in C | 3 | 0 | 0 | 1BPOPL107 | C Programming Lab | 0 | 0 | 2 |
| 1BEBT105 | Elements of Biotechnology and Biomimetics | 3 | 0 | 0 | 1BEBTL107 | Elements of Biotechnology Lab | 0 | 0 | 2 |
| 1BSSA105 | Principles of Soil Science and Agronomy | 3 | 0 | 0 | 1BSSAL107 | Soil Science and Agronomy Field Lab | 0 | 0 | 2 |
| 1BEAE105 | Elements of Aeronautica Engineering | 3 | 0 | 0 | 1BEAEL107 | Elements of Aeronautica Engineering Lab | 0 | 0 | 2 |
| 1BECHE105 | Elements of Chemical Engineering | 3 | 0 | 0 | 1BECHEL107 | Elements of Chemical Engineering Lab | 0 | 0 | 2 |

**Integrated courses (IC), combining theory with practical components.**

(i) Theory sessions shall be conducted for 3 hours per week, while the practical sessions shall be conducted for 2 hours per week.

(ii)Theory components shall be evaluated through both Continuous Internal Evaluation (CIE) and Semester End Examination (SEE).

(iii)The practical component shall be assessed only through CIE.

**The Mathematics/Physics courses shall be taught by a single faculty member per session, with no sharing of the course (subject) modules.**

The **tutorial sessions** for the **mathematics course** shall be conducted in the laboratory environment using **Maxima/Mathematica/ Python/Scilab/MATLAB software** to enhance computational understanding and application skills **(**one hour for problem solving and one hour laboratory session).

All students admitted to the engineering program have to complete **Applied Mathematics-I and Applied Mathematics-II** in I and II semesters by selecting the courses prescribed for

their stream, viz. CV, ME, EEE or CSE, under the heading Mathematics –I and Mathematics-II.
Those who have completed the physics course under the heading Applied Physics in I semester have to select the prescribed stream wise chemistry course under the heading Applied chemistry during II semester.

**Programme Specific Courses (PSC):** Programme Specific Courses (PSC) are a set of core courses tailored to a specific branch or discipline of engineering in which a student is enrolled (e.g., Mechanical Engineering, Computer Science, Civil Engineering, etc.). These courses are intended to provide students with in-depth knowledge and specialized skills essential for professional competence in the chosen field. Students must select and complete the course from this group that **corresponds to their admitted program stream.** Similarly, students are also required to choose and pass laboratory courses that are specific to their stream from the **Programme Specific Courses Laboratory (PSCL) group.**

**Engineering Sciences Courses-I(ESC-I):** These courses are designed to broaden the technical knowledge of students beyond their core area of study. These courses enable students to gain a foundational understanding of engineering principles from other stream courses. Students are required to select and complete two courses that are not belong to their admitted program stream. For example, a student admitted to the any programme of the Civil Engineering stream should not select Introduction to Building Sciences but any other two. One course shall be selected under ESC-I and another course under ESC-II. The two courses must be different from the other.

**Computer-Aided Engineering Drawing:** The courses under this category are stream-specific. Students must select and complete the course that corresponds to their admitted engineering stream**.**

The **Student Induction Programme** (SIP), initiated by the All India Council for Technical Education (AICTE), is designed to help newly admitted students in technical institutions transition smoothly into the higher education environment. It aims to familiarize students with the institutional culture, foster connections with peers and faculty, and provide a foundation for holistic learning. Activities under SIP may include Physical Activities, Creative Arts, Universal Human Values, Literary Events, Proficiency Modules. Lectures shall be by Eminent Personalities, Local Area Visits, Department/Branch Familiarization, and Innovation-related sessions.
The first year of the Engineering programmes is composed of I semester, II semester and Summer Semester. SIP activities shall be scheduled in the afternoon sessions during the first week of class commencement of I and II semesters only.
The specific programmes to be conducted will be notified separately by the University via the academic calendar or through a separate notification.

**AICTE Activity Points Requirement for BE/B.Tech. Programmes**
As per AICTE guidelines (refer Chapter 6 – *AICTE Activity Point Program, Model Internship Guidelines*), in addition to academic requirements, students must earn a specified number of **Activity Points** to be earned is to be eligible for the award of their degree.

- **Regular students** admitted to a 4-year degree program must earn **100 Activity Points**.
- **Lateral entry students** (joining from the second year) must earn **75 Activity Points**.
- **Students transferred** from other universities directly into the fifth semester must earn **50 Activity Points** from the date of entry into VTU.

These Activity Points are **non-credit** and will not be considered for **the SGPA/CGPA** or be used for **vertical progression**. However, they are mandatory for the **award of the degree**, and the points earned will be reflected on the **eighth semester Grade Card**.
The hours spent for earning the activity points shall not be counted for regular attendance requirements. Students can accumulate these points at any time during their program, including on weekends, holidays, and vacations starting from the year of admission, provided they meet the minimum hours of engagement prescribed for each activity.
If a student fails to earn the required Activity Points, the eighth-semester Grade Card will be withheld until the requirement is fulfilled. Consequently, the degree will be awarded only after the Grade Card has been released.

| Sl No | Stream | UG Programmes under the stream with code |
|---|---|---|
| 1 | Civil Engineering Stream (CV) | (1) Civil engineering (CV), (2) Mining Engineering (MI) |
| 2 | Mechanical Engineering Stream (ME) | (1)Aeronautical Engineering (AE), (2)Aerospace Engineering (AS),(3) Agricultural Engineering (AG),(4)Automation and Robotics (AR), (5)Automobile Engineering (AU), (6)Chemical Engineering (CH), (7) Industrial & Production Engineering (IP), (8)Industrial Engineering & Management (IM), (9) Manufacturing Science and Engineering (MS), (10) Marine Engineering (MR), (11) Mechanical & Smart Manufacturing (MM), (12) Mechanical Engineering (ME), (13)Mechatronics (MT), (14) Petrochem Engineering (PC), (15)Robotics & Automation (RA),(16) Robotics and Artificial Intelligence (RI),(17)Silk Technology (ST), (18) Textile Technology (TX),(19)Energy Engineering (ER),(20) Smart Agritech (SA). |
| 3 | Electrical and Electronics Engineering Stream (EEE) | (1)Electronics & Communication Engineering (EC), (2)Biomedical Engineering (BM), (3)Electrical & Electronics Engineering (EE), (4) Electronics & Instrumentation Engineering (EI),(5) Electronics & Telecommunication Engineering (ET),(6) Industrial IoT (IO), (7) Medical Electronics Engineering (ML),(8) Electronics Engineering (VLSI Design and Technology) (VL),(9) Electronics & Communication( Advanced Communication Technology) (EA),(10) Electronics & Computer Engineering (UE). |
| 4 | Computer Science and Engineering Stream (CSE) | (1) Computer Science and Engineering (CS), (2) Computer Engineering (CE), (3)Artificial Intelligence and Data Science (AD), (4) Artificial Intelligence and Machine Learning (AI),(5) Biotechnology (BT),(6)Computer & Communication Engineering (CM), (7) Computer Science and Business System (CB),(8) Computer Science and Design (CG),(9) Computer Science and Engineering (IoT) (CO), (10)CSE(Artificial Intelligence and Machine Learning) (CI),(11) CSE(Artificial Intelligence) (CA),(12) CSE(Cyber Security) (CY), (13)CSE(Data Science) (CD),(14) CSE(IoT and Cyber Security including Block Chain Technology) (IC), (15) Data Science (DS), (16) Information Science & Engineering (IS),(17) Computer Science (CR). |
| | | |

**Scheme of Teaching and Examinations (2025)**
Outcome-Based Education (OBE)and Choice-Based Credit System (CBCS)
(Effective from the academic year 2025-26)

**II Semester (For the students who have studied Physics group in I semester)**

| Sl. No | Course and Course Code | | Course Title | TD/PSB | Teaching Hours/Week | | | | Examination | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Theory Lecture | Tutorial | Practical/ Drawing | SAAE | Duration in hours | CIE Marks | SEE Marks | Total Marks | |
| | | | | | L | T | P | S | | | | | |
| 1 | ASC | 1BMATx201 | Applied Mathematics -II (Stream Specific) | Maths Dept | 3 | 2 | 0 | | 03 | 50 | 50 | 100 | 04 |
| 2 | ASC(IC) | 1BCHEx202 | Applied Chemistry (Stream Specific) | CHE Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 3 | ETC | 1BAIA203/ BETC205x | Introduction to AI and Applications | Any Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 4 | ESC | 1BESC204x | Engineering Science Course-II | Respective Engg Dept | 3 | 0 | 0 | | 03 | 50 | 50 | 100 | 03 |
| 5 | PLC(IC) | 1BPLC205x | Programming Language Course | CSE & allied Dept | 3 | 0 | 2 | | 03 | 50 | 50 | 100 | 04 |
| 6 | AEC | 1BENG206 | Communication Skills | Humanities Dept | 1 | 0 | 0 | | 02 | 50 | 50 | 100 | 01 |
| 7 | AEC (NCMC) | 1BICO207 | Indian Constitution & Engineering Ethics | Humanities Dept | 1 | 0 | 0 | | 01 | 100 | 0 | 100 | PP |
| 8 | AEC/SDC | 1BPRJ258 | Interdisciplinary Project-Based Learning | Respective Dept (Multiple Dept) | 0 | 0 | 0 | 2 | 02 | 50 | 50 | 100 | 01 |
| | | | | **TOTAL** | **17** | **02** | **05** | **02** | **20** | **450** | **350** | **800** | **20** |

**ASC**-Applied Science Course, **IC** – Integrated Course (Practical Course Integrated with Theory Course), **ESC**- Engineering Science Courses, **PLC(IC)**- Programming Language Course (Integrated Course), **AEC**- Ability Enhancement Course, **NCMC**: Non Credit Mandatory Course, **TD/PSB**- Teaching Department / Paper Setting Board, **HSMC**- Humanity, Social Science and management Course, **S- (SAAE)-** Students' Academic Activity Engagement Hours, **AEC/SDC**- Ability Enhancement Course/Skill Development course, **CIE** –Continuous Internal Evaluation, **SEE**- Semester End Examination, **PP** : (Pass/Pass) is assigned to a noncredit course. "PP" represents pass in course provided students have successfully completed the CIE requirements. Otherwise, "NP-not pass shall be awarded. "PP" is essential for the award of the degree

**Integrated courses (IC), combining theory with practical components.**
The theory sessions shall be conducted for 3 hours per week, while the practical sessions shall be conducted for 2 hours per week.

- The theory component will be evaluated through both Continuous Internal Evaluation (CIE) and Semester End Examination (SEE).
- The practical component will be assessed only through CIE.

**Communication Skills:** This course shall be conducted in a laboratory environment

| Applied Mathematics-II | | | | | Applied Chemistry | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Title** | **L** | **T** | **P** | **Code** | **Title** | **L** | **T** | **P** |
| 1BMATC201 | Differential Calculus and Numerical Methods: CV Stream | 3 | 2 | 0 | 1BCHEC202 | Applied Chemistry for Sustainable Structure & Material Design (CV) | 3 | 0 | 2 |
| 1BMATM201 | Multivariable Calculus and Numerical Methods: ME Stream | 3 | 2 | 0 | 1BCHEM202 | **Applied Chemistry for Advanced Metal Protection and Sustainable Energy Systems** (ME) | 3 | 0 | 2 |
| 1BMATE201 | Calculus, Laplace Transform And Numerical Techniques: EEE stream | 3 | 2 | 0 | 1BCHEE202 | **Applied Chemistry for Emerging Electronics and Futuristic Devices** (EEE, ECE) | 3 | 0 | 2 |
| 1BMATS201 | Numerical Methods: CSE Stream | 3 | 2 | 0 | 1BCHES202 | Applied Chemistry for Smart Systems (CSE) | 3 | 0 | 2 |
| **Engineering Sciences Courses II(ESC-II)** | | | | | **Programming Language Courses (PLC)** | | | | |
| 1BESC204A | Building Sciences & Mechanics | 3 | 0 | 0 | 1BPLC205E | Introduction to C Programming (for non-IT programmes) | 3 | 0 | 2 |
| 1BESC204B | Introduction to Electrical Engineering | 3 | 0 | 0 | 1BPLC205B | Python Programming (For CSE and allied programmes) | 3 | 0 | 2 |
| 1BESC204C | Introduction to Electronics & Communication Engineering | 3 | 0 | 0 | | | | | |
| 1BESC204D | Introduction to Mechanical Engineering | 3 | 0 | 0 | | | | | |
| 1BESC204E | Essentials of Information Technology | 3 | 0 | 0 | | | | | |

**The Mathematics/Chemistry courses shall be taught by a single faculty member per session, with no sharing of the course (subject) modules.** The **tutorial sessions** for the **mathematics course** shall be conducted in the laboratory environment using **Maxima/Mathematica/ Python/Scilab/MATLAB software** to enhance computational understanding and application skills.

Students admitted to a specific engineering stream are required to select and successfully complete **Applied Mathematics-II** and **Applied Chemistry courses** that are aligned to their program stream.

**Engineering Sciences Courses-II(ESC-II):** These courses are designed to broaden the technical knowledge of students beyond their core area of study. These courses enable students to gain a foundational understanding of engineering principles from other disciplines. Students are required to select and complete a course under ESC-II that does not belong to their admitted program stream. Students should select a course other than that was selected under ESC-I and other than course not belonging to their stream.

**F**or the course ***Interdisciplinary Project*** **(BPRJ259)**, it is mandatory to form a team comprising students from multiple engineering disciplines. For example, a project team may include students from Mechanical Engineering, Electronics and Communication Engineering (ECE), and Computer Science and Engineering (CSE), working collaboratively to design and implement the project.

| PYTHON PROGRAMMING | | Semester | I/II |
|---|---|---|---|
| Course Code | **1BPLC105B/205B** | CIE Marks | 50 |
| Teaching Hours/Week (L:T:P: S) | 3:0:2:0 | SEE Marks | 50 |
| Total Hours of Pedagogy (Theory and Lab hours) | 40 + 24 (Practical) | Total Marks | 100 |
| Credits | 4 | Exam Hours | 3 |
| Examination type (SEE) | Theory | | |

**Course outcome (Course Skill Set)**

At the end of the course, the student will be able to:

CO1: Develop scripts using primitive language constructs of python.
CO2: Identify the methods to manipulate primitive python data structures.
CO3: Make use of Python standard libraries for programming.
CO4: Build scripts for performing file operations.
CO5: Illustrate the concepts of Object-Oriented Programming as used in Python.

**Module-1**

**The way of the program:** The Python programming language, what is a program? What is debugging? Syntax errors, Runtime errors, Semantic errors, Experimental debugging.

**Variables, Expressions and Statements:** Values and data types, Variables, Variable names and keywords, Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator.

**Iteration:** Assignment, Updating variables, the for loop, the while statement, The Collatz 3n + 1 sequence, tables, two-dimensional tables, break statement, continue statement, paired data, Nested Loops for Nested Data.

**Functions:** Functions with arguments and return values.

**Chapters: 1.1-1.7, 2.1-2.12, 3.3, 4.4, 4.5**

Number of Hours:8

**Module-2**

**Strings:** Working with strings as single things, working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, the in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method.

**Tuples:** Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.

**Lists:** List values, accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices.

**Chapter: 5.1, 5.2, 5.3**

Number of Hours: 8

**Module-3**

**Dictionaries:** Dictionary operations, dictionary methods, aliasing and copying.

**Numpy:** About, Shape, Slicing, masking, Broadcasting, dtype.

**Files:** About files, writing our first file, reading a file line-at-a-time, turning a file into a list of lines, Reading the whole file at once, working with binary files, Directories, fetching something from the Web.

**Chapter: 5.4, 6.1-6.5, 7.1-7.8**

Number of Hours:8

**Module-4**

**Modules:** Random numbers, the time module, the math module, creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot Operator, Three import statement variants.
**Mutable versus immutable and aliasing**
**Object oriented programming:** Classes and Objects — The Basics, Attributes, Adding methods to our class, Instances as arguments and parameters, Converting an instance to a string, Instances as return values.

**Chapter: 8.1-8.8, 9.1, 11.1**

Number of Hours: 8

**Module-5**

**Object oriented programming**: Objects are mutable, Sameness, Copying.
**Inheritance:** Pure functions ,Modifiers, Generalization, Operator Overloading, Polymorphism.
**Exceptions**: Catching Exceptions, Raising your own exceptions.

**Chapter: 11.2.2-11.2.4, 11.3.2-11.3.9, 12.1, 12.2**

Number of Hours:8

## PRACTICAL COMPONENTS OF IPCC

**PART – A: FIXED SET OF EXPERIMENTS**

1. a. Develop a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide).
   b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.
2. a. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.
   b. Write a python program to create a list and perform the following operations
   - Inserting an element
   - Removing an element
   - Appending an element
   - Displaying the length of the list
   - Popping an element
   - Clearing the list
3. a. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.
   b. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with a suitable message.
4. Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use a dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display the dictionary slice of the first 10 items.
5. Develop a program to read 6 subject marks from the keyboard for a student. Generate a report that displays the marks from the highest to the lowest score attained by the student. [Read the marks into a 1-Dimesional array and sort using the Bubble Sort technique].
6. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()].

7. Develop a function named DivExp which takes TWO parameters a, b, and returns a value c (c=a/b). Write a suitable assertion for a>0 in the function DivExp and raise an exception for when b=0. Develop a suitable program that reads two console values and calls the function DivExp.
8. Define a function that takes TWO objects representing complex numbers and returns a new complex number with the sum of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers.
9. Text Analysis Tool: Build a tool that analyses a paragraph: frequency of each word, longest word, number of sentences, etc.
10. Develop Data Summary Generator: Read a CSV file (like COVID data or weather stats), convert to dictionary form, and allow the user to run summary queries: max, min, average by column.
11. Develop Student Grade Tracker: Accept multiple students' names and marks. Store them in a list of tuples or dictionaries. Display summary reports (average, topper, etc.).
12. Develop a program to display contents of a folder recursively (Directory) having sub-folders and files (name and type).

**Suggested Learning Resources: (Text Book/ Reference Book/ Manuals):**

**Text books:**

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers- How to think like a computer scientist: learning with python 3. Green Tea Press, Wellesley, Massachusetts,2020 https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf

**Reference books / Manuals:**

1. Al Sweigart," Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners",2nd Edition, No Starch Press, 2022. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Kyla McMullen, Elizabeth Matthews and June Jamrich Parsons, Programming with Python, Cengage, 2023.

**Web links and Video Lectures (e-Resources):**

https://www.learnbyexample.org/python/

https://www.learnpython.org/

https://pythontutor.com/visualize.html#mode=edit

**Teaching-Learning Process (Innovative Delivery Methods):**

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Chalk and talk
2. PPT presentation
3. Demonstration
4. Problem-Based Learning (PBL)
5. Case-Based Teaching

**Assessment Structure:**

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying **50% weightage** (i.e., 50 marks each).
The CIE Theory component will be 25 marks and CIE Practical component will be 25 marks.

The CIE Theory component consists of IA tests for 25 marks and Continuous Comprehensive Assessments (CCA) for 5 marks. The CIE Practical component for continuous assessments will be for 15 marks through rubrics and for lab tests will be for 5 marks.

- To qualify and become eligible to appear for SEE, in the **CIE theory component**, a student must score at least **40% of 30 marks**, i.e., **12 marks**.
- To qualify and become eligible to appear for SEE, in the **CIE Practical component**, a student must secure **a minimum of 40% of 20 marks**, i.e., **08 marks**.
- To pass the **SEE**, a student must secure **a minimum of 35% of 50 marks**, i.e., **18 marks**.
- A student is deemed to have **successfully completed the course** if the **combined total of CIE and SEE is at least 40 out of 100 marks**.

**Continuous Comprehensive Assessments (CCA):**

CCA will be conducted for a total of 5 marks. It is recommended to include any one learning activity aimed at enhancing the holistic development of students. This activity should align with course outcomes and promote higher-order thinking and application-based learning.

**Learning Activity -1: (Marks- 5)**

Students must identify a real-life scenario and develop a Python-based solution using fundamental programming constructs/Data structures (Below given are the sample examples).

1. E.g.: Banking System: Simulate bank accounts using classes. Implement deposit, withdraw, and balance check using class methods. Create your own utility module.
2. E-commerce Cart System: Build a class Product, extend it with Electronics, Clothing using inheritance. Create a Cart class. Handle errors like invalid quantity using custom exceptions.
3. Smart Attendance System: Use file I/O to maintain logs, dictionaries for student info, and exception handling for invalid entries.
4. Develop/Simulate snake and ladder game by choosing suitable data structures of Python.

**CIE Practical component:**

The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report using a defined set of rubrics. Each experiment report can be evaluated for 30 marks. The summation of all the experiments marks to be scaled down to 15 marks.

The laboratory test (duration 03 hours) at the end of the last week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 5 marks. For laboratory test, the student is required to conduct one experiment each from both Part A and Part B.

**Rubrics for Learning Activity (Based on the nature of learning activity, design the rubrics for each activity):**

Note: Marks obtained (25) is scaled down to 5.

| Component & CO-PO Mapping | Outstanding (5) | Exceeds Expectations (4) | Meets Expectations (3) | Needs Improvement (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Identification of real-life problem and its relevance [CO1] [PO2] | Clearly defined and contextually relevant problem; innovative approach | Relevant and well-described problem | Partially relevant with limited context | Vague or not fully relevant problem | No identifiable or valid problem |
| Use of primitive constructs (variables, loops, functions, conditionals) [CO1] [PO1] | All constructs used correctly with proper logic and flow | Most constructs used properly | Basic constructs applied with some errors | Minimal construct usage with logical flaws | Incorrect or missing constructs |
| Manipulation of Python data structures (lists, tuples, dictionaries, sets) [CO2] [PO1] | Effective and optimized usage of Data Structures | Mostly appropriate usage | Some usage with basic understanding | Incorrect or limited use | Not used or misused entirely |
| Use of standard libraries and file operations (if applicable) [CO3, CO4] [PO5] | Libraries and file operations used correctly and meaningfully | Minor issues in usage | Limited or partially correct use | Attempted but faulty implementation | Not attempted or irrelevant |
| Code structure, modularity, and documentation [CO4] [PO9, PO11] | Modular, structured code with comments and output samples | Structured code with basic documentation | Limited comments or unclear structure | Poor documentation and readability | No documentation, disorganized code |

**Rubrics for CIE – Continuous assessment:**

| Component & CO-PO Mapping | Outstanding (5) | Exceeds Expectations (4) | Meets Expectations (3) | Needs Improvement (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Fundamental Knowledge: Understanding the problem statement<br><br>[CO1-5]<br>[PO1, PO2] | The student has in depth knowledge of the topics related to the problem. Student is able to completely understand the problem definition. | Student has good knowledge of some of the topics related to problem. Student is able to understand the problem definition. | Student is capable of narrating the answer but not capable to show in depth knowledge and the problem definition. | Student has not understood the concepts partially. Student is able to partially understand the problem definition | Student has not understood the concepts and the problem definition clearly. |
| Design of algorithm/flow chart and program<br><br>[CO1-5]<br>[PO2, PO3] | Student is capable of discussing more than one design for his/her problem statement and capable of proving the best suitable design with proper reason. | Student is capable of discussing few designs for his/her problem statement but not capable of selecting best. | Student is capable of discussing single design with its merits and de-merits. | Student is capable of explaining the design. | Student is capable of explaining the design partially. |
| Implementation (Program coding) with suitable tools<br><br>[CO1-5]<br>[PO5, PO8] | Student is capable of implementing the design with best suitable language structure considering optimal solution/optimal efficiency. | Student is capable of implementing the design with best suitable language structure and should be capable of explaining it. | Student is capable of implementing the design with proper explanation. | Student is capable of implementing the design. | Student is capable of implementing the design with errors. |
| Program debugging and testing with suitable tools<br><br>[CO1-5]<br>[PO5, PO8] | Student is capable to compile and debug the program with no errors (syntax, semantic and logical). | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with full understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with partial understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with no understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with assistance. |
| Results & interpretation /analysis<br><br>[CO1-5]<br>[PO4] | Student is able to run the program on various cases and compare the result with proper analysis. | Student is able to run the program for all the cases. | Student is able to run the code for few cases and analyze the result. | Student is able to run the program but not able to analyze the result. | Student is able to run the program but not able to verify the correctness of the result. |
| Demonstration and documentation<br><br>[CO1-4]<br>[PO8, PO9, PO11] | Demonstration and lab record is well-organized, with clear sections. | Demonstration and lab record is organized, with clear sections, but some | Demonstration and lab record lacks clear organization or structure. Some sections are | Demonstration and lab record is poorly organized, with missing or unclear sections. | Demonstration and lab record is poorly organized, with missing sections. Record |

| | | | | | |
|---|---|---|---|---|---|
| | The record is well structured with suitable formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | sections are not well-defined. The record is structured with formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | unclear or incomplete. The record is partially structured with formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | The record is not properly structured with suitable formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | not submitted on time. The record is not structured with minimum formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). |

**Rubrics for CIE Test:**

| Component & CO-PO Mapping | Excellent (5) | Good (4) | Fair (3) | Marginal (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Fundamental Knowledge (2)<br><br>[CO1, CO2] [PO1] | The student has well depth knowledge of the topics related to the problem & course | Student has good knowledge of some of the topics related to problem & course | Student has average knowledge of some of the topics related to problem & course | Student is capable of narrating the answer but not capable to show in depth knowledge | Student has not understood the concepts clearly |
| Understanding of problem definition (1)<br><br>[CO1, CO2] [PO2] | Student is able to completely understand the problem definition | Student is able to understand the problem definition but not clearly | Student has a basic understanding of the problem definition that is partial or superficial | Student is able to Shows minimal or unclear understanding of the problem definition | Student is not able to understand the problem definition |
| Design and Implementation (3)<br><br>[CO1, CO2] [PO3] | Student is capable of design and implementing with best suitable construct for the given problem definition | Student is capable of design and implementing with some construct for the given problem definition | Student is capable of design and implementing the core part of the construct for the given problem definition | Student is partially capable of design and implementing with some algorithm for the given problem definition | Student is not capable of design and implementing |
| Result & Analysis (2)<br><br>[CO1, CO2] [PO4] | Student is able to run the program on various data inputs and compare the result with proper inference. | Student will be able to run the program on various data inputs and fair knowledge in comparing the result with proper inference | Student will be able to run the code for few data/datasets and analyze the output. | Student will be able to run the code for few data inputs but not analyze the output. | Student will be not able to run the program and not able to analyze the result. |
| Communication (Viva voce) (2)<br><br>[CO3] [PO8, PO9] | Good Verbal & nonverbal communication skills with precise and correct terminologies/ answers. | Good verbal Communication skills with precise and correct terminologies/ answers. | Average Communication but with precise and correct terminologies/ answers. | Average Communication but with imprecise and incorrect terminologies/ answers | Poor Communication (Minimal interaction/answers) |

| INTRODUCTION TO C PROGRAMMING | | Semester | I/II |
|---|---|---|---|
| Course Code | **1BPLC205E/105E** | CIE Marks | 50 |
| Teaching Hours/Week (L:T:P: S) | 3:02:0 | SEE Marks | 50 |
| Total Hours of Pedagogy (Theory and Lab hours) | 40 + 24 (Practical) | Total Marks | 100 |
| Credits | 4 | Exam Hours | 3 |
| Examination type (SEE) | Theory | | |

**Course outcomes (Course Skill Set)**

At the end of the course, the student will be able to:

CO1: Explain the fundamental structure of a C program and primitive constructs.
CO2: Apply decision-making and iterative control structures to solve simple computational problems.
CO3: Develop programs using arrays and string operations to solve real-world problems.
CO4: Construct user-defined functions to modularize the solution to the given problems.
CO5: Build programs using structures and pointers for complex data representation and access.

**Module-1**

**Flowchart and Algorithms:** Art of Programming through Algorithms & Flowcharts.
**Overview of C:** History of C, Importance of C, Basic Structure of C Programs, Programming Style, Compiling and Executing a 'C' Program.
**Constants, Variables and Data Types:** Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, Data Types, Declaration of Variables, Assigning Values to Variables, Defining Symbolic Constants, Declaring a Variables as Constants and Volatile, Input/Output Statements in C.

**Textbook: Chapter 1. 6, 2.1, 2.2, 2.8, 2.9, 2.10, Chapter 3.2 to 3.14, Chapter 5.1 to 5.5**

Number of Hours: 8

**Module-2**

**Operators:** Introduction to Operators, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Precedence of Arithmetic Operators.
**Decision Making, Branching, Looping:** Introduction, Decision Making with IF Statement, Simple IF Statement, The IF..ELSE Statement, Nesting of IF..ELSE Statements, The ELSE IF Ladder, The Switch Statement, The ?: Operator, The GOTO Statement, WHILE, DO, FOR, Jumps in LOOPS.

**Textbook: Chapter 4.1 to 4.7, 4.12, Chapter 6.1 to 6.9, Chapter 7.1 to 7.5** Number of Hours: 8

**Module-3**

**Arrays and Strings:** Introduction, Declaration and Initialization of One-dimensional and Two-Dimensional Arrays, Declaring and Initializing String Variables, Example programs using arrays ,Reading Strings from Terminal, Writing Strings to Screen, Arithmetic Operations on Characters, Comparison of Two Strings, String-handling Functions.

**Textbook: Chapter 8.1 to 8.6, Chapter 9.2 to 9.5, 9.7, 9.8** Number of Hours: 8

**Module-4**

**User-defined Functions:** Introduction, Need for User-defined Functions, A Multi-functional Program, Elements of User-defined Functions, Definition of Function, Return Values and their Types, Function Calls, Function Declaration, No Arguments and no Return Values, Arguments but no Return Values, Nesting of Functions.

**Textbook: Chapter 10.1 to 10.8, 10.10 to 10.14**

Number of Hours:8

**Module-5**

**Structures and Pointers:** Introduction, Defining a Structure, Declaring and Accessing Structure Variables and Members, Structure Initialization, Copying and Comparing Structure Variables, Array of Structures, Arrays within Structures.
**Pointers:** Introduction, Understanding Pointers, Accessing the Address of Variable, Declaring pointer variables, initialization of pointers, accessing variables through its pointer.

**Textbook: Chapter 11.1 to 11.6, 11.8, 11.19, Chapter 12.1 to 12.6**

Number of Hours:8

**PRACTICAL COMPONENT OF IPCC**

1. Develop a program to calculate the temperature converter from degree to Fahrenheit.
2. Develop a program to find the roots of quadratic equations.
3. Develop a program to find whether a given number is prime or not.
4. Develop a program to find key elements in an array using linear search.
5. Given age and gender of a person, develop a program to categorise senior citizen (male & female).
6. Generate Floyd's triangle for given rows.
7. Develop a program to find the transpose of a matrix.
8. Develop a program to concatenate two strings, find length of a string and copy one string to other using string operations.
9. Develop a modular program to find GCD and LCM of given numbers.
10. Develop a program to declare the structure of employees and display the employee records with higher salary among two employees.
11. Develop a program to add two numbers using the pointers to the variables**.**
12. Develop a program to find the sum of digits of a given number.
13. Develop a program to perform Matrix Multiplication.
14. Develop a program to create an array of structures to store book details and check whether a specific book, as requested by the user, is available or not.

**Suggested Learning Resources: (Textbook/ Reference Book/ Manuals):**

**Textbooks:**

1. Programming in ANSI C, 9e, E Balaguruswamy, Tata McGraw Hill Education.

**Reference books / Manuals:**

1. PROGRAMMING IN C, Reema Thareja, Oxford University, Third Edition, 2023**.**
2. The 'C' Programming Language, Brian W. Kernighan and Dennis M. Ritchie, Second Edition, Prentice Hall of India, 2015

**Web links and Video Lectures (e-Resources):**

1. elearning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html

2. https://nptel.ac.in/courses/106/105/106105171/ MOOC

Courses can be adopted for more clarity in understanding the topics and verities of problem-solving methods.

- https://www.tutorialspoint.com/what-is-an-algorithm-and-flowchart-in-c-language
- https://www.tutorialspoint.com/cprogramming/c_data_types.htm
- https://www.tutorialspoint.com/cprogramming/c_operators.htm
- https://www.ccbp.in/blog/articles/decision-making-statements-in-c
- https://www.tutorialspoint.com/cprogramming/c_arrays.htm
- https://www.geeksforgeeks.org/variables-in-c/
- https://www.w3schools.com/c/c_arrays.php
- https://www.programiz.com/c-programming/c-strings
- https://www.programiz.com/c-programming/c-pointers
- https://www.scaler.com/topics/c/structures-c/

**Teaching-Learning Process (Innovative Delivery Methods):**

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Flipped Classroom
2. Problem-Based Learning (PBL)
3. Case-Based Teaching
4. Simulation and Virtual Labs
5. ICT-Enabled Teaching

**Assessment Structure:**

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying **50% weightage** (i.e., 50 marks each).
The CIE Theory component will be 30marks and CIE Practical component will be 20 marks.

The CIE Theory component consists of IA tests for 25 marks and Continuous Comprehensive Assessments (CCA) for 5 marks. The CIE Practical component for continuous assessments will be for 15 marks through rubrics and for lab tests will be for 5 marks.

- To qualify and become eligible to appear for SEE, in the **CIE theory component**, a student must score at least **40% of 30 marks**, i.e., **12 marks**.
- To qualify and become eligible to appear for SEE, in the **CIE Practical component**, a student must secure **a minimum of 40% of 20 marks**, i.e., **08 marks**.
- To pass the **SEE**, a student must secure **a minimum of 35% of 50 marks**, i.e., **18 marks**.
- A student is deemed to have **successfully completed the course** if the **combined total of CIE and SEE is at least 40 out of 100 marks**.

**Continuous Comprehensive Assessments (CCA):**

CCA will be conducted for a total of 5 marks. It is recommended to include any, one learning activity aimed at enhancing the holistic development of students. This activity should align with course outcomes and promote higher-order thinking and application-based learning.

**Learning Activity -1: Programming Assignment (Marks- 5)**

INSTRUCTIONS:

1. Course instructor will refer to HackerRank or any other platform to derive the questions for problem-solving.
2. Course Instructor must identify programming problems from these sections: Statements (control), Arrays, Strings, Structures & Unions and Functions.
3. Courser instructor will assign question ONE from each section to the students for design of algorithm/flowchart, program and coding/execution.
4. Students must demonstrate the solutions to the course instructor and submit the record containing algorithm/flowchart, program, debugging/execution and results with observations.
5. Course instructor must evaluate the student performance as per the rubrics.

**Rubrics for Learning Activity (Based on the nature of learning activity, Develop the rubrics for each activity):**

Note: Marks obtained (25) is scaled down to 5.

**Rubrics for Learning Activity:**

| Component & CO-PO Mapping | Outstanding (5) | Exceeds Expectations (4) | Meets Expectations (3) | Needs Improvement (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Clarity & Simplicity of algorithm/program [CO1] [PO9] | Algorithm/Programs are self-explanatory, specific, and well-structured for the intended activity; no ambiguity is present. | Programs are clear and mostly specific; minor ambiguity is present. | Programs are somewhat clear but could be more specific; moderate ambiguity. | Programs are vague and lack clarity; high ambiguity. | Programs are unclear, incomplete, or irrelevant to the activity. |
| Appropriate Use of language constructs and design of algorithm/program [CO2-5] [PO1, PO3] | Demonstrates precise and creative usage of the language construct and structured programming | Correctly applies the language construct with minor gaps or missed opportunities. | Uses the language construct, but with partial understanding or inconsistent usage. | Limited understanding of the language construct; incorrect or weak usage. | No evidence of correct/relevant language construct use. |
| Compilation, Debugging, Analysis & Comparison of Results for various cases. [CO2-5] [PO2, PO4, PO5] | Provides clear and correct results with analysis for multiple cases; comparisons among cases highlight key strengths and weaknesses. | Provides correct results with analysis for multiple cases, though slightly less detailed. | Provides correct results with limited analysis; comparisons are present but shallow. | Provides correct results. Minimal analysis: comparisons are weak or incomplete. | Results are partially correct. No meaningful analysis or comparison. |
| Creativity, efficiency of | Demonstrates outstanding | Demonstrates creativity and | Shows moderate creativity; | Minimal creativity: | No creativity or problem- |

| | | | | | |
|---|---|---|---|---|---|
| Problem-Solving/program [CO2-5] [PO3, PO11] | creativity and innovation in writing programs, especially for problem-solving or design tasks. | some innovation; Program solutions are practical. | programs are functional but not innovative. | programs are repetitive or unimaginative. | solving/Programming is evident. |
| Documentation & Reflection [CO1-5] [PO8/PO9/PO11] | Documentation is complete, well-organized, and includes deep reflection on improvements across iterations. | Documentation is complete with some reflection on program refinement. | Documentation is present but lacks detail or depth in reflection. | Incomplete documentation; reflection is minimal. | No documentation or reflection provided as per schedule. |

**Rubrics for CIE – Continuous assessment:**

| Component & CO-PO Mapping | Outstanding (5) | Exceeds Expectations (4) | Meets Expectations (3) | Needs Improvement (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Fundamental Knowledge: Understanding the problem statement<br><br>[CO1]<br>[PO1, PO2] | The student has in depth knowledge of the topics related to the problem. Student is able to completely understand the problem definition. | Student has good knowledge of some of the topics related to problem. Student is able to understand the problem definition. | Student is capable of narrating the answer but not capable to show in depth knowledge and the problem definition. | Student has not understood the concepts partially. Student is able to partially understand the problem definition | Student has not understood the concepts and the problem definition clearly. |
| Design of algorithm/flow chart and program<br><br>[CO2-5]<br>[PO2, PO3] | Student is capable of discussing more than one design for his/her problem statement and capable of proving the best suitable design with proper reason. | Student is capable of discussing few designs for his/her problem statement but not capable of selecting best. | Student is capable of discussing single design with its merits and de-merits. | Student is capable of explaining the design. | Student is capable of explaining the design partially. |
| Implementation (Program coding) with suitable tools<br><br>[CO2-5]<br>[PO5, PO8] | Student is capable of implementing the design with best suitable language structure considering optimal solution/optimal efficiency. | Student is capable of implementing the design with best suitable language structure and should be capable of explaining it. | Student is capable of implementing the design with proper explanation. | Student is capable of implementing the design. | Student is capable of implementing the design with errors. |

| | | | | | |
|---|---|---|---|---|---|
| Program debugging and testing with suitable tools<br><br>[CO2-5]<br>[PO5, PO8] | Student is capable to compile and debug the program with no errors (syntax, semantic and logical). | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with full understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with partial understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with no understanding of error descriptions. | Student is able to compile and debug the program with errors (syntax, semantic and logical) and rectified errors with assistance. |
| Results & interpretation /analysis<br><br>[CO1-5]<br>[PO4] | Student is able to run the program on various cases and compare the result with proper analysis. | Student is able to run the program for all the cases. | Student is able to run the code for few cases and analyze the result. | Student is able to run the program but not able to analyze the result. | Student is able to run the program but not able to verify the correctness of the result. |
| Demonstration and documentation<br><br>[CO1-5]<br>[PO8, PO9, PO11] | Demonstration and lab record is well-organized, with clear sections.<br>The record is well structured with suitable formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | Demonstration and lab record is organized, with clear sections, but some sections are not well-defined.<br>The record is structured with formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | Demonstration and lab record lacks clear organization or structure. Some sections are unclear or incomplete.<br>The record is partially structured with formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | Demonstration and lab record is poorly organized, with missing or unclear sections.<br>The record is not properly structured with suitable formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). | Demonstration and lab record is poorly organized, with missing sections. Record not submitted on time.<br>The record is not structured with minimum formatting (e.g: font, spacing, labelling of figures and tables, equations numbered and etc). |

**Rubrics for CIE Test:**

| Component & CO-PO Mapping | Excellent (5) | Good (4) | Fair (3) | Marginal (2) | Unsatisfactory (1) |
|---|---|---|---|---|---|
| Fundamental Knowledge (2)<br><br>[CO1]<br>[PO1] | The student has well depth knowledge of the topics related to the problem & course | Student has good knowledge of some of the topics related to problem & course | Student has average knowledge of some of the topics related to problem & course | Student is capable of narrating the answer but not capable to show in depth knowledge | Student has not understood the concepts clearly |
| Understanding of problem definition (1)<br><br>[CO2+-5]<br>[PO2] | Student is able to completely understand the problem definition | Student is able to understand the problem definition but not clearly | Student has a basic understanding of the problem definition that is partial or superficial | Student is able to Shows minimal or unclear understanding of the problem definition | Student is not able to understand the problem definition |
| Design and Implementation (3)<br><br>[CO2-5]<br>[PO3] | Student is capable of design and implementing with best suitable construct for the given problem definition | Student is capable of design and implementing with some construct for the given problem definition | Student is capable of design and implementing the core part of the construct for the given problem definition | Student is partially capable of design and implementing with some algorithm for the given problem definition | Student is not capable of design and implementing |
| Result & Analysis (2)<br><br>[CO2-5]<br>[PO4] | Student is able to run the program on various data inputs and compare the result with proper inference. | Student will be able to run the program on various data inputs and fair knowledge in comparing the result with proper inference | Student will be able to run the code for few data/datasets and analyze the output. | Student will be able to run the code for few data inputs but not analyze the output. | Student will be not able to run the program and not able to analyze the result. |
| Communication (Viva voce) (2)<br><br>[CO1-5]<br>[PO8, PO9] | Good Verbal & nonverbal communication skills with precise and correct terminologies/ answers. | Good verbal Communication skills with precise and correct terminologies/ answers. | Average Communication but with precise and correct terminologies/ answers. | Average Communication but with imprecise and incorrect terminologies/ answers | Poor Communication (Minimal interaction/answers) |