COURSE PACK

# 1. THE SCHEME

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Title** | **Programming Skills with Computational Maths** | | | **Course Type** | | | **Lab** | | | |
| **Course Code** | **307** | | | **Class** | | | **B-Tech Core and All specialization (II-YR)** | | | |
| **Instruction delivery** | **Activity** | **Credits** | **Credit Hours** | **Total Number of Classes per Semester** | | | | | **Assessment in Weightage** | |
| **Lecture** | 0 | 0 |
| **Tutorial** | 0 | 0 | **Theory** | **Tutorial** | **Practical** | | **Self-study** | **LAB TEST** | **LAB EXAM** |
| **Practical** | 2 | 4 |
| **Self-study** | 0 | 0 |
| **Total** | 2 | 4 | 0 | 0 | 2 | | 0 | 25 | 25 |
| **Course Lead** | Dr. Nidhi Agarwal | | **Course Coordinator** | Dr. Sandeep Kumar M | | | | | | |
| **Names Course Instructors** | **Practical** | | | **Practical** | | | | | | |
| 1. Abhishek Kumar Pandey  2. Akash Raghuvanshi  3. Akhilesh Kumar Tripathi  4. AMIT KUMAR  5. ANKIT VERMA  6. Arvind Dagur  7. ASHWANI KUMAR  8. Basu Dev Shivahare  9. Brijesh Kumar Singh  10. D SALANGAI NAYAGI  11. Deepa Joshi  12. Devendra Kumar  13. Dr. Pragya Srivastava  14. Dr. S. Prabaharan  15. Dr. Sonia Kukreja  16. Dr. Sunil Kumar  17. Dr.Sandeep Kumar M  18. Durgesh Nandini  19. Garima Verma  20. Geeta Gayatri Behera  21. Hradesh kumar  22. Indervati  23. Ishana Attri  24. Jagveer Singh  25. Jawed Akhtar  26. K.suresh  27. KARPAGA SELVI S  28. KIRTI  29. Mahesh Kumar Chouhan  30. MANDEEP  31. Manish Verma  32. Manmohan Singh  33. Mukesh Kumar | | | 34. Mullapudi Navyasri  35. Muniraj Gupta  36. Murari Krishna Saha  37. NEERAJ KUMAR  38. Neha Bagwari  39. NIDHI AGARWAL  40. Nidhi Sharma  41. P.SELVRAJ  42. Piyoush Kumar  43. Pradeep Chauhan  44. Pragya  45. Ms.Pragya Tewari  46. Dr.Prakash Anand  47. Dr.PRASHANT DIXIT  48. Ms.R. Indrakumari  49. R.RADHIKA  50. R.Sathiya priya  51. Radha Rani  52. RAJIV CHOPRA  53. Rakesh Bharati  54. Rikendra  55. Rikendra  56. Riman Mandal  57. Ruchi Sharma  58. S P RAMESH  59. Sandeep Bhatia  60. Savita Kumari  61. Simarpreet Singh  62. Mr. Satyam Raj  63. Alok Kumar  64. C. Jayaprakash  65. Chandira Prakash N  66. S. Syed Sabir Mohamed | | | | | | |

# 2. COURSE OVERVIEW

This lab course equips students with the ability to design and implement efficient mathematical algorithms drawn from algebra, number theory, and modular arithmetic, with a strong focus on competitive programming applications. It covers advanced techniques such as binary exponentiation, GCD algorithms, linear Diophantine equations, primality testing, and number-theoretic functions. Students will also explore bit manipulation, number systems, and arbitrary-precision arithmetic to handle computations beyond standard data type limits. By combining mathematical theory with hands-on coding practice, the course strengthens analytical and problem-solving skills essential for technical interviews, system-level challenges, and algorithmic competitions.

# 3. COURSE OBJECTIVES

* Develop the ability to design and implement efficient algorithms for computational problems in algebra and number theory.
* Equip students with techniques for handling large numerical computations, including binary exponentiation and arbitrary-precision arithmetic.
* Strengthen understanding of modular arithmetic concepts such as modular inverses, Chinese Remainder Theorem, and modular exponentiation.
* Introduce advanced GCD algorithms and their applications, including solving linear Diophantine equations under various constraints.
* Familiarize students with prime number generation, primality testing, and integer factorization methods used in competitive programming.
* Enhance proficiency in bit manipulation, number systems, and combinatorial enumeration techniques.
* Integrate mathematical theory with practical coding to prepare students for technical interviews and algorithmic competitions.
* Cultivate precision, efficiency, and analytical skills necessary for systems-level programming and large-scale computational tasks.

# 4. PREREQUISITE COURSE

|  |  |  |
| --- | --- | --- |
| PREREQUISITE COURSE REQUIRED | No |  |
| If, yes please fill in the Details | Course code | Course Title |
| NA | NA |

1. **PROGRAM OUTCOMES (POs):**

|  |  |
| --- | --- |
| **PO No.** | **Description of the Program Outcome** |
| **PO1** | Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems. |
| **PO2** | Problem Analysis: Identify, formulate, review research literature and analyse complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4). |
| **PO3** | Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5). |
| **PO4** | Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8). |
| **PO5** | Modern Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6). |
| **PO6** | The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7). |
| **PO7** | Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9). |
| **PO8** | Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams. |
| **PO9** | Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences. |
| **PO10** | Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one’s own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.. |
| **PO11** | Life-Long Learning: Recognize the need for, and have the preparation and ability for:  i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8). |

1. **PROGRAM SPECIFIC OUTCOMES (PSOs):**

Program Specific Outcomes (PSO) are statements that describe what the graduates of a discipline-specific program should be able to do. Two to Three PSOs per program should be designed.

|  |  |
| --- | --- |
| **PO No.** | **Description of the Program-Specific Outcome** |
| **PSO1** | Have the ability to work with emerging technologies in Computer Science and Engineering requisite to Industry 4.0. |
| **PSO2** | Demonstrate Engineering Practice learned through industry internship and research project to solve live problems in various domains. |

# 7. COURSE CONTENT (LAB)

|  |
| --- |
| **CONTENT (Syllabus)** |
| LAB:  **Fundamentals: Binary Exponentiation**: Effective computation of large exponents modulo a number, Effective computation of Fibonacci numbers, Applying a permutation k times; Euclidean algorithm for computing the greatest common divisor: Least common multiple, Binary GCD, Extended Euclidean Algorithm;  **Linear Diophantine Equation:** Finding the number of solutions and the solutions in a given interval, Generating all possible solutions, Find the solution with minimum value of  x + y ; Fibonacci Numbers: Fibonacci Coding, Formulas for the nth Fibonacci number, Closed-form expression, Fibonacci in linear time, Periodicity modulo p.  **Prime Numbers:** Sieve of Eratosthenes, Linear Sieve, Primality tests, Integer factorization; Number Theoretic Functions: Number of Divisors; Modular Arithmetic: Modular Multiplicative Inverse, Linear Congruence Equation, Chinese Remainder Theorem.  **Number System:** Balanced Ternary. Gray code; Bit manipulation: Binary Number, Bit operators, Shift operators, Submask Enumeration; Arbitrary-Precision Arithmetic: Classical Integer Long Arithmetic. |

## 8. COURSE OUTCOMES (COs)

After the completion of the course, the student will be able to:

|  |  |
| --- | --- |
| **CO No.** | **Description of the Course Outcome** |
| 307.1 | Design and implement efficient mathematical algorithms used in competitive programming and systems-level problem solving, particularly from algebra and number theory. |
| 307.2 | Handle large numerical computations using techniques like binary exponentiation and arbitrary-precision arithmetic, enabling them to work beyond the limitations of standard data types. |
| 307.3 | Strengthen problem-solving skills necessary for competitive programming, technical interviews, and advanced algorithmic challenges by combining mathematical theory with practical coding. Identify and formulate research problem in their interesting domain. |

## 9. TAXONOMY LEVEL OF THE COURSE OUTCOMES

### Mapping of COs with Bloom’s Level

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CO No. | Remember **KL1** | Understand **KL 2** | Apply  **KL 3** | Analyse  **KL 4** | Evaluate  **KL 5** | Create  **KL 6** |
| 307.1 |  | √ | √ | √ |  |  |
| 307.2 |  |  |  |  | √ | √ |
| 307.3 |  |  |  | √ | √ | √ |

## 10. COURSE ARTICULATION MATRIX

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **COs#/ POs** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** |
| 307.1 | - | 2 | - | - | - | 2 | - | - | - | 1 | - | - | - |
| 307.2 | - | 2 | 1 | - | - | - | 3 | - | - | - | - | 2 | 2 |
| 307.3 | 2 | - | - | - | 3 | 1 | - | 2 | 1 | - | 1 | 2 | 2 |

**Note: 1-**Low, 2-Medium, 3-High \ \*first semester first course and first Course Outcome

# 11. TYPICAL EXAMPLE OF COURSES, CREDIT HOURS AND TEACHING HOURS

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type of Course** |  | **Credits Hours** | | |  | **Hours of engagement/ Week** | | | | | **12 weeks/ semester** | **Remarks** |
| **Theory** | **Tutorial** | **Practical** | **Self-study** | **Total** | **Theory** | **Tutorial** | **Practical** | **Self-study** | **Total** | **Total no. of classes** |
| Lab Course | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 4 | 12 | 12 classes for lab |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **L-No** | **Topic for Delivery** | **Tutorial / Practical Plan** | **Skill** | **Competency** |
| 1 | Sum of first N natural numbers | Practical | Design and implement efficient mathematical algorithms and programming concepts | CO1 |
| 2 | Factorial modulo 10^9+7 | Practical |
| 3 | Count trailing zeroes in n! | Practical |
| 4 | Check if a number is power of 2 | Practical |
| 5 | Count set bits in a number | Practical |
| 6 | Binary to Decimal and vice versa | Practical |
| 7 | Floor and ceil of square root | Practical |
| 8 | Multiply two large numbers as strings | Practical | Handle numerical computations using suitable programming and logical techniques | CO2 |
| 9 | Power of a number (recursive and iterative) | Practical |
| 10 | Find all divisors of a number | Practical |
| 11 | Divisibility by 9 using digit sum | Practical |
| 12 | GCD of two numbers / strings | Practical |
| 13 | GCD of array of numbers | Practical |
| 14 | LCM of array | Practical |
| 15 | Count numbers coprime to N | Practical | Strengthening problem-solving skills necessary for competitive programming, and advanced algorithmic challenges. | CO3 |
| 16 | Sum of all divisors from 1 to N | Practical |
| 17 | Check if number is prime (6k ± 1) | Practical |
| 18 | Sieve of Eratosthenes | Practical |
| 19 | Count primes between two numbers | Practical |
| 20 | Smallest Prime Factor (SPF) | Practical |
| 21 | Modular multiplication | Practical |
| 22 | Binary Exponentiation (a^b % m) | Practical |
| 23 | Modular inverse (Fermat Little Theorem) | Practical |
| 24 | Modular inverse (Extended Euclid Algorithm) | Practical |
| 25 | Compute nCr % p (using Fermat’s Theorem) | Practical |
| 26 | Modular exponentiation with large power | Practical |
| 27 | Chinese Remainder Theorem | Practical |
| 28 | Linear Congruence ax ≡ b mod m | Practical |
| 29 | Euler's Totient for all numbers | Practical |
| 30 | Check if two numbers are coprime | Practical |

## 12. BIBLIOGRAPHY

* **Text Book:**

1. The Art Of Computer Programming, Volume 1 Fundamental Algorithms Third Edition by Donald E. Knuth.

2. Introduction to Algorithms, Third Edition by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Reference Books:

1. Competitive Programmer’s Handbook by Antti Laaksonen.
2. Hacker's Delight, Second Edition by Henry S. Warren, Jr.
3. Concrete Mathematics, Second Edition by Ronald L. Graham, Donald E. Knuth & Oren Patashnik.
4. Programming Challenges: The Programming Contest Training Manual by Steven S. Skiena & Miguel A. Revilla Cooper Donald R, Schindler Pamela S and Sharma JK, “Business Research Methods”, Tata McGraw Hill Education, 11e (2012).

## 13. COURSE ASSESSMENT

Assessment forms an integral part of curriculum design. A learning-teaching system can only be effective if the student’s learning is measured at various stages which means while the student processes learning (Assessment for Learning) a given content and after completely learning a defined content (Assessment of Learning). Assessment for learning is referred to as formative assessment, that is, an assessment designed to inform instruction.

The ability to use and apply the knowledge in different ways may not be the focus of the assessment. With regard to designing assessments, the faculty members must be willing to put in the time required to create a valid, reliable assessment, that ideally would allow students to demonstrate their understanding of the information while remaining. The following are the five main areas that assessment reporting should cover.

1. **Learning Outcomes**: At the completion of a program, students are expected to know their knowledge, skills, and attitude. Depending on whether it is a UG or PG program, the level of sophistication may be different. There should be no strict rule on the number of outcomes to be achieved, but the list should be reasonable, and well-organized.
2. **Assessable Outcomes**: After a given learning activity, the statements should specify what students can do to demonstrate. Criteria for demonstration are usually addressed in rubrics and there should be specific examples of work that doesn’t meet expectations, meets expectations, and exceeds expectations. One of the main challenges is faculty communication whether all faculty agreed on explicit criteria for assessing each outcome. This can be a difficult accomplishment when multiple sections of a course are taught or different faculty members. Hence there is a need for common understanding among the faculty on what is assessed and how it is assessed.
3. **Assessment Alignment**: This design of an assessment is sometimes in the form of a curriculum map, which can be created in something as easy as an Excel spreadsheet. Courses should be examined to see which program outcomes they support, and if the outcome is assessed within the course. After completion, program outcomes should be mapped to multiple courses within the program.
4. **Assessment Planning**: Faculty members need to have a specific plan in place for assessing each outcome. Outcomes don’t need to be assessed every year, but faculty should plan to review the assessment data over a reasonable period of time and develop a course of action if the outcome is not being met.
5. **Student Experience**: Students in a program should be fully aware of the expectations of the program. The program outcomes are aligned on the syllabus so that students are aware of what course outcomes they are required to meet, and how the program outcomes are supported. Assessment documents should clearly communicate what is being done with the data results and how it is contributing to the improvement of the program and curriculum.

**Designing quality assessment tools** or tasks involves multiple considerations if it is to be fit for purpose. The set of assessments in a course should be planned to provide students with the opportunity to learn as they engage with formative tasks as well as the opportunity to demonstrate their learning through summative tasks. Encouraging the student through the use of realistic, authentic experiences is an exciting challenge for the course faculty team, who are responsible for the review and quality enhancements to assessment practices.

## 14. FORMATIVE AND SUMMATIVE ASSESSMENT

**Assessment Pattern for Theory Course:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of Course (T)** |  | **CIE** | **Total Marks** | | **Final Marks** |
| LAB TEST | LAB EXAM | LAB TEST | LAB EXAM |
| LAB | 25 | 25 | 25 | 25 | 50 |