AC – 20/04/2024
Item No. – 6.7 Sem. II (2a)

As Per NEP 2020

University of Mumbai

Syllabus for Basket of Minor	
Board of Studies in Information Technology	
UG First Year Programme	
Semester	**II**
Title of Paper	**Credits 2/ 4**
I. IT_Problem Solving using Computer(Minor)	**2**
II.	
From the Academic Year	**2024-2025**

Name of the Course: IT_Problem Solving Using Computer

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course, Problem Solving Using Computer, is designed to teach students how to solve complex problems using computational thinking, logical and algorithmic thinking, and effective problem-solving strategies. Students will learn how to anticipate and deal with errors while testing and debugging their code to ensure optimal functionality. The course also places an emphasis on evaluation, as students will be taught how to evaluate their solutions for correctness, efficiency, elegance, usability, and trade-offs. Module 2 of this course introduces Python and its basic types, operations, and functions. It also covers more advanced constructs such as program state and tkinter, organizing code using modules and packages, using abstractions and patterns to simplify programming, and effective modelling. Students will also learn how to test and evaluate their programs, verifying and validating them in parts and as a whole.
2	**Vertical :**	Minor
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To introduce students to computational thinking and how it is used to solve problems using a computer. CO 2. To develop logical and algorithmic thinking skills in students, with an emphasis on identifying and avoiding common mistakes. CO 3. To equip students with the necessary problem-solving skills to tackle real-world challenges by breaking down the problem, devising a solution and implementing an effective strategy. CO 4. To teach students how to use abstraction to simplify complex problems and how to create models to make predictions and enhance understanding. CO 5. To prepare students for the reality of dealing with errors by teaching them how to anticipate, detect and mitigate program bugs through effective testing and debugging techniques.	

8	**Course Outcomes (OC):** OC 1. Students will be able to analyze a given problem and select an appropriate computational strategy to solve it. OC 2. Students will be able to apply logical and algorithmic thinking to develop error-free computer programs that solve complex problems. OC 3. Students will be able to decompose a problem into smaller components and devise a solution strategy, effectively using patterns and generalization to design an effective solution. OC 4. Students will be able to use abstraction to simplify complex problems and effectively create models that simulate real-world scenarios. OC 5. Students will be able to anticipate and handle errors by implementing testing and debugging procedures while also efficiently evaluating their own solutions with respect to their correctness, efficiency, and usability.
9	**Modules:-** **Module 1: 15 hours**
	1. **Computational Thinking:** Objectives, What is it? How is it used? Disclaimers 2. **Logical and Algorithmic thinking:** Objectives, Approach, Logical thinking, Algorithmic thinking, 'Gotchas' 3. **Problem Solving and Decomposition:** Objectives, Where to start? Defining a problem, devising solution, decomposition, other effective strategies, patterns and generalization 4. **Abstraction and Modelling:** Objectives, Abstraction, Modelling 5. **Anticipating and dealing with errors:** Objectives, coming to terms with bugs, Designing out the bugs, Mitigating errors, Testing, Debugging, deciding which errors to fix 6. **Evaluation a solution:** Objectives, Solution evaluation, Is it correct? Is it efficient? Is it elegant? Is it usable? Trade-offs
	Module 2: (15 hours)
	7. **Introducing Python:** Objectives, Introduction, First steps, Basic types, Basic operations, Function, Comments 8. **Effective Building Blocks:** Logic, Basic arithmetic constructs, Program state, Advanced constructs 9. **Organising Code:** Objectives, tkinter, Separating concerns, defining information scope, using modules, packages 10. **Using abstractions and patterns:** Objectives, finding patterns in programs, abstractions in programming, built-in types, creating own types, ready-made patterns 11. **Effective modelling:** Objectives, entities, relationships, processes, usage, modelling tips 12. **Testing and evaluation programs:** Objectives, introduction, anticipating bugs, verification and validation, testing in parts, testing the whole, debugging

	13. **Example**	
10	**Text and References:** 1. COMPUTATIONAL THINKING: A beginner's guide to problem-solving and programming, Karl Beecher, BCS Learning & Development Ltd, 2017 2. Problem Solving with Computers, Greg W. Scragg, Jones and Barlett Publishers, 1997	
11	**In 10 above**	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology