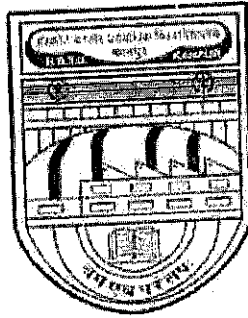


**Course Curriculum and Detailed Syllabi
for
B. Tech. 1st Year Computer Science and Engineering
(Common to Information Technology and CSE – AI/ML)**



Academic Session: 2026-27

Department of Computer Science & Engineering

Harcourt Butler Technical University

Kanpur (208002)

As per NEP 2020

[Handwritten signatures and initials]

Course Structure and Evaluation Scheme (School of Engineering)- 2026-27

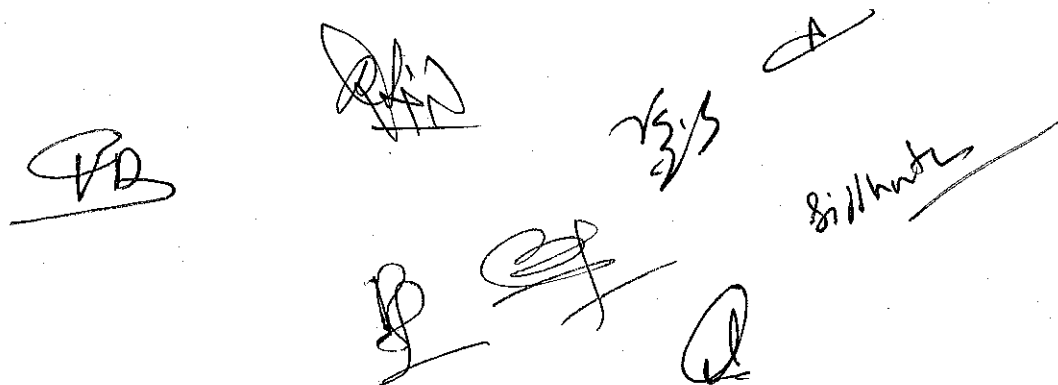
I B.Tech. Computer Science and Engineering

Evaluation Scheme: Semester I

S. No.	Course Type	Subject Code	Course title	Credits (L-T-P)	Sessional Marks				ESM	Total Marks
					MSE 1	TA	Lab	Total		
1	ESC	DCS101	Programming for Problem Solving	4(3-1-0)	30	20	-	50	50	100

Evaluation Scheme: Semester II

S. No.	Course Type	Subject Code	Course title	Credits (L-T-P)	Sessional Marks				ESM	Total Marks
					MSE 1	TA	Lab	Total		
1	PCC	DCS201	Python Programming	3(2-0-2)	15	20	15	50	50	100

The bottom section of the page contains several handwritten signatures and initials. On the left, there is a signature that appears to be 'V.D.'. In the center, there are two more signatures, one above the other. To the right of these, there are initials 'V.S./S' and a signature that looks like 'Siddhanta'. At the top right, there is a small signature that looks like 'A'.

Subject Name: **PROGRAMMING FOR PROBLEM SOLVING (PPS)**

Subject Code: **DCS101**

Credits [L-T-P]: **4[3-1-0]**

Prerequisite

Course Objectives

The objective of this course is to develop computational thinking and problem-solving abilities among students through systematic algorithm design and programming techniques. The course enables students to analyse engineering problems, design logical solutions, and implement efficient programs using structured programming concepts.

Course Outcomes

After completing the course, students will be able to:

CO	Outcome	
CO1	Analyze the problem and think in terms of solution design.	K2,K3,K4
CO2	Use the appropriate control statements of C language to solve the given problem.	K3
CO3	Implement different Operations on arrays and use functions to solve the given problem using C.	K3, K4
CO4	Understand pointers and designing of basic data structures like stacks, queues in C.	K2, K3 K4
CO5	Make use of files and file operations to store and retrieve data and build simple problem-solving applications using C programming.	K2, K3,K5

K1- Remember, K2- Understand, K3-Apply, K4-Analyse, K5- Evaluate, K6- Create

Course Articulation Matrix (CO-PO-PSO Matrix)

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	-	2	-	-	-	-	-	1	2	1	2	1
CO2	3	3	2	-	2	-	-	-	-	-	1	3	3	2	1
CO3	3	3	2	-	2	-	-	-	-	-	2	3	3	2	1
CO4	3	3	3	1	2	-	-	-	-	-	2	3	3	2	2
CO5	3	3	3	1	2	-	-	2	2	1	3	3	3	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High); If there is no correlation, put "-".

Program Outcomes

1. **Engineering Knowledge:** Apply core fundamentals (math, science, and engineering) to complex engineering problems.
2. **Problem Analysis:** Identify and formulate complex engineering problems and substantiate conclusions.
3. **Design/Development of Solutions:** Design systems/components that meet specific public health, safety, and environmental needs.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and methods to synthesize information.
5. **Modern Tool Usage:** Select and apply appropriate techniques, resources, and modern engineering IT tools.
6. **The Engineer and the World:** Understand the impact of professional engineering solutions in societal, environmental, and global contexts.
7. **Ethics:** Apply professional and ethical responsibilities, including inclusive practices and equity.
8. **Individual and Collaborative Teamwork:** Function effectively as an individual, and as a member or leader in diverse teams.
9. **Communication:** Comprehend, write reports, and give effective presentations.
10. **Project Management and Finance:** Apply engineering management and financial principles to multidisciplinary environments.
11. **Lifelong Learning:** Recognize the need for and have the preparation and ability to engage in independent and lifelong learning.

Program Specific Outcomes

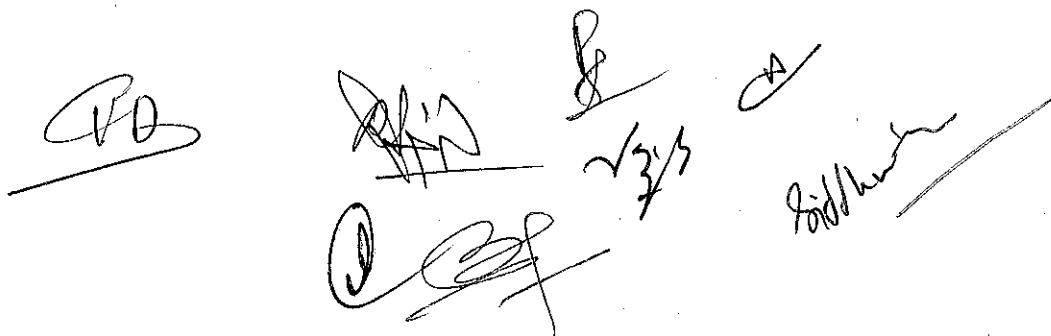
By the completion of the B. Tech. Computer Science & Engineering program, the students will achieve the following program specific outcomes:

PSO 1: The ability to understand, analyse and develop applications in the field of algorithms, system software, databases, web design, networking and artificial intelligence.

PSO 2: The ability to apply standard practices and strategies in software project development using suitable programming environments to deliver a quality product.

PSO 3: The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

PSO 4: The ability to use research-based knowledge to do literature surveys, formulate problems, design and carry out experimentation, analyse and interpret experimental results for complex research problems.

The bottom of the page contains several handwritten signatures and initials. From left to right, there is a signature that appears to be 'PD', a signature that appears to be 'RIN', a signature that appears to be 'S', a signature that appears to be 'A', and a signature that appears to be 'Siddharth'.

Course Contents

Unit 1

Introduction to computational thinking, Problem identification and analysis, Decomposition of problems, Abstraction and algorithmic thinking, Flowcharts and pseudocode, Algorithms to programs: specification, top-down development and stepwise refinement, Introduction to programming environment, use of high level programming language for the systematic development of programs. Introduction to the design and implementation of correct, efficient and maintainable programs, Structured Programming, Trace an algorithm to depict the logic.

Unit 2: C Programming Language

Operators: Unary operators, Arithmetic & logical operators, Bit wise operators, Assignment operators and expressions, Conditional expressions, Precedence and order of evaluation.

Decision Making and Iterative Problem Solving: if-else, switch, break, and continue, the comma operator, goto statement, for, while, do-while.

Unit 3

Functions: built-in and user-defined, function declaration, definition and function call, and parameter passing: call by value, call by reference, recursive functions, Multi-file programs.

Array: Linear arrays, multidimensional arrays, passing arrays to functions, Arrays and strings.

Unit 4

Pointers: value at (*) and address of (&) operator, pointer to pointer, Dynamic Memory Allocation, calloc and malloc functions, array of pointers, function of pointers, structures and pointers.

Data Organization: Introduction to Trees, Graphs, Stacks and queues, Searching techniques, Sorting techniques, Introduction to complexity awareness

Unit 5

File Handling: Opening and closing a data file, creating a data file, read and write functions, unformatted data files. structures for record handling, Mini-project development, Testing and validation, Introduction to collaborative programming.

Handwritten signatures and initials at the bottom of the page, including 'VD', 'R', 'V/S', 'A', 'Siddharth', and others.

Text Books

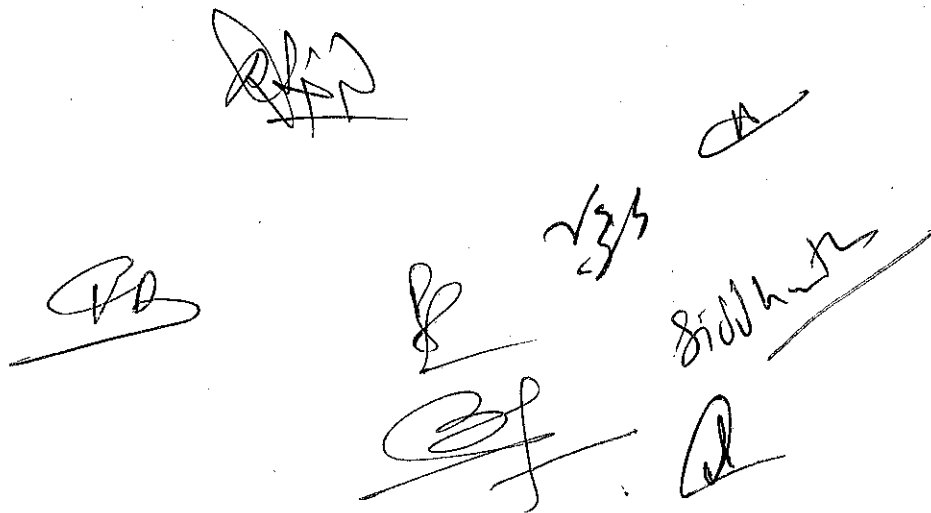
1. Yashavant Kanetkar, Let Us C, BPB Publications.
2. Reema Thareja, Programming in C, Oxford University Press.
3. E. Balagurusamy, Programming in ANSI C, McGraw Hill.
4. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Pearson Education.

Reference Books

1. Byron Gottfried, Programming with C, Schaum Series, McGraw Hill.
2. Ashok N. Kamthane, Programming with ANSI and Turbo C, Pearson Education.
3. Herbert Schildt, C: The Complete Reference, McGraw Hill.
4. R.G. Dromey, How to Solve It by Computer, Pearson Education.
5. V. Rajaraman, Computer Programming in C, PHI Learning.

Online Resources

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs53
2. <https://www.coursera.org/learn/computational-thinking-problem-solving>
3. <https://pll.harvard.edu/course/cs50-introduction-computer-science>

The bottom half of the page contains several handwritten signatures and initials. At the top center is a signature that appears to be 'RAN'. Below it and to the left is a signature 'PD'. To the right of 'PD' is a signature 'P' with '2/3/3' written above it. Further right is a signature 'Siddhant' with a horizontal line underneath. At the bottom center is a large signature 'Sf'. To the right of 'Sf' is a signature 'Ad'.

Subject Name: PYTHON PROGRAMMING

Subject Code: DCS201

Credits [L-T-P]: 3[2-0-2]

Course Objectives

The course is designed to understand the fundamentals of Python Programming and acquire skills of implementing control structure, string, advanced functions like iteration and recursion and object-oriented programming concepts in Python.

Course Outcomes

After completing the course, students will be able to:

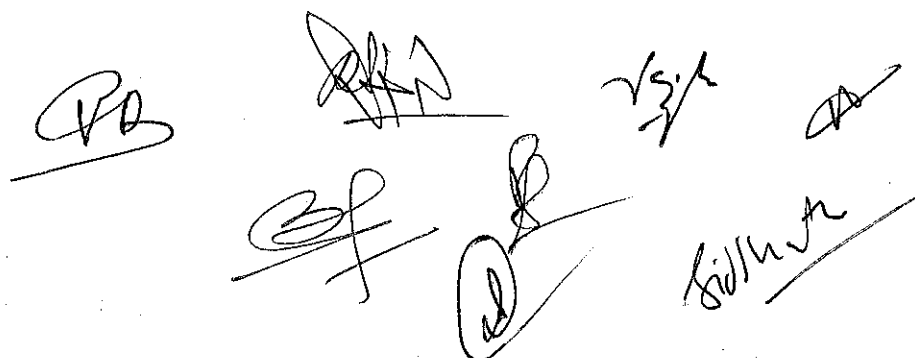
CO	Outcome	
CO1	Understanding Fundamentals of Python Programming	K1, K2,K3
CO2	Apply and implement Control Structures.	K2,K3,K4
CO3	Design implements Strings and Functions in Python.	K2,K3,K4
CO4	Evaluate advanced functions like iteration and recursion.	K3,K4
CO5	Understand object Oriented Programming concepts in Python	K4,K5

K1- Remember, K2- Understand, K3-Apply, K4-Analyse, K5- Evaluate, K6- Create

Course Articulation Matrix (CO-PO-PSO Matrix)

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO2	PSO 3	PSO 4
CO1	3	3	3	-	-	-	-	2	-	-	2	3	2	1	1
CO2	3	3	3	-	-	-	-	2	-	-	2	3	2	1	1
CO3	3	3	1	-	-	-	-	2	-	-	2	3	2	1	1
CO4	3	3	3	-	-	-	-	2	-	-	2	3	2	1	1
CO5	3	3	3	-	-	-	-	2	-	-	2	2	1	-	-

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High); If there is no correlation, put "-".



Program Outcomes

1. **Engineering Knowledge:** Apply core fundamentals (math, science, and engineering) to complex engineering problems.
2. **Problem Analysis:** Identify and formulate complex engineering problems and substantiate conclusions.
3. **Design/Development of Solutions:** Design systems/components that meet specific public health, safety, and environmental needs.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and methods to synthesize information.
5. **Modern Tool Usage:** Select and apply appropriate techniques, resources, and modern engineering IT tools.
6. **The Engineer and the World:** Understand the impact of professional engineering solutions in societal, environmental, and global contexts.
7. **Ethics:** Apply professional and ethical responsibilities, including inclusive practices and equity.
8. **Individual and Collaborative Teamwork:** Function effectively as an individual, and as a member or leader in diverse teams.
9. **Communication:** Comprehend, write reports, and give effective presentations.
10. **Project Management and Finance:** Apply engineering management and financial principles to multidisciplinary environments.
11. **Lifelong Learning:** Recognize the need for—and have the preparation and ability to engage in—independent and lifelong learning.

Program Specific Outcomes

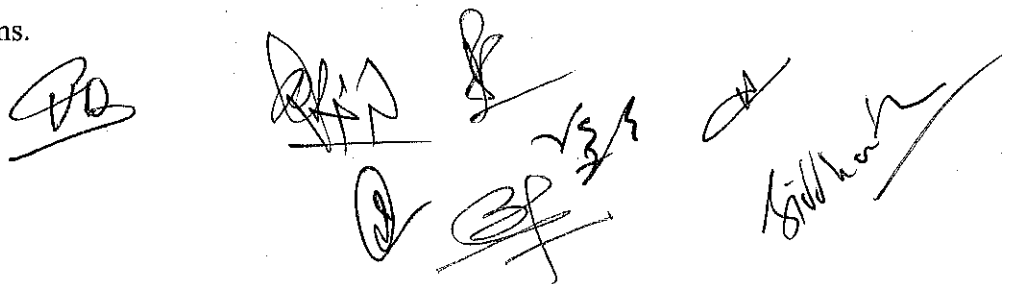
By the completion of the B. Tech. Computer Science & Engineering program, the students will achieve the following program specific outcomes:

PSO 1: The ability to understand, analyse and develop applications in the field of algorithms, system software, databases, web design, networking and artificial intelligence.

PSO 2: The ability to apply standard practices and strategies in software project development using suitable programming environments to deliver a quality product.

PSO 3: The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

PSO 4: The ability to use research-based knowledge to do literature surveys, formulate problems, design and carry out experimentation, analyse and interpret experimental results for complex research problems.

A collection of handwritten signatures in black ink, likely representing the approval of the program outcomes by faculty members. The signatures are scattered across the bottom right of the page.

Prerequisite: Programming for Problem Solving

Course Contents

Unit 1: Introduction

The Programming Cycle for Python, Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression.

Unit 2: Conditionals and Loops

Conditional statement in Python: if-else statement, its working and execution, Nested-if statement and Else if statement in Python, Expression Evaluation & Float Representation, Loops: Purpose and working of loops, while loop including its working, For Loop, Nested Loops, Break and Continue.

Unit 3: Strings and Functions

Strings: Length of the string, Concatenation and Repeat operations, Indexing and Slicing of Strings. Python Data Structure: Tuples, Unpacking Sequences, Lists, Mutable Sequences, List Comprehension, Sets, Dictionaries, Functions: Parts of a Function, Execution of a Function, Keyword and Default Arguments, Scope Rules, Higher Order Functions: Treat functions as first-class Objects, Lambda Expressions.

Unit 4: Classes and Files

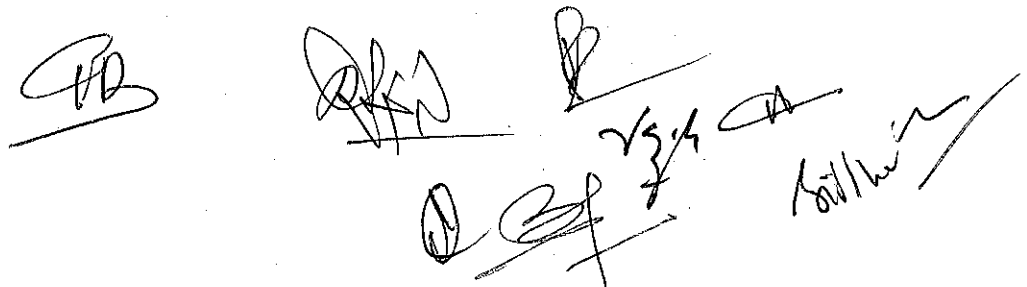
Generate prime numbers with the help of Sieve of Eratosthenes algorithm, File I/O: File input and output operations in Python Programming Exceptions and Assertions Modules: Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming, Classes: Definition and operations in the classes, Special Methods (such as `__init__`, `__str__`, comparison methods and Arithmetic methods etc.), Class Example, Inheritance, Inheritance and OOP.

Unit 5: Iterators & Recursion

Recursive Fibonacci, Tower of Hanoi, Search: Simple Search, Binary Search, Estimating Search Time in Simple Search and Binary Search, Sorting & Merging: Selection Sort, Merge List, Merge Sort, Higher Order Sort.

Lab Work

1. To develop Python programs with conditionals and loops.
2. To define Python functions and to use Python data structures -- lists, tuples, dictionaries
3. To do input/output with files in Python
4. Write a python program to implement prime numbers with the help of Sieve of Eratosthenes algorithm
5. Write a python program to implement exception handling.
6. Write a Python Program to implement inheritance.
7. Write a Python Program to perform Linear Search
8. Write a Python Program to perform Binary Search



Handwritten signatures and initials are present at the bottom of the page, including a large signature on the left, a signature in the center, and several initials and marks on the right, including one that appears to say 'V.S.' and another that says 'Sillwin'.

9. Write a Python Program to perform selection sort
10. Write a Python Program to perform insertion sort.
11. Write a Python Program to perform merge sort.

Text Books

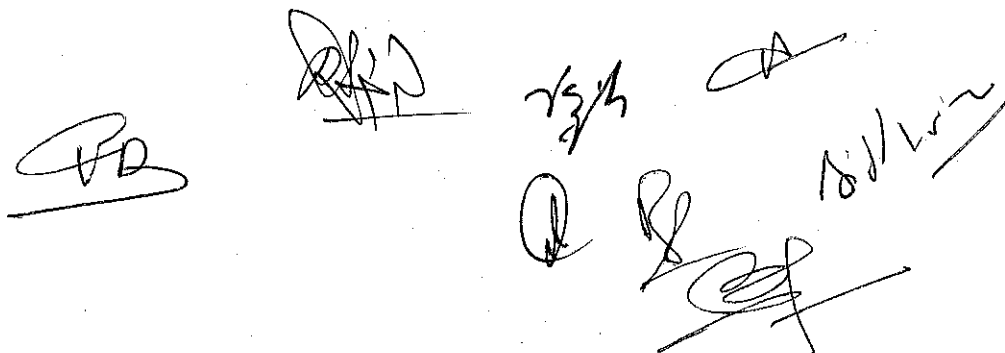
1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist Updated for Python 3, Shroff O'Reilly Publishers, 2nd edition 2016.
2. Guido van Rossum and Fred L. Drake Jr, An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011
3. John V Guttag, "Introduction to Computation and Programming Using Python", Revised and expanded Edition, MIT Press, 2013.

Reference Books

1. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 1st edition 2016.
2. Timothy A. Budd, Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2nd edition 2015.
3. Kenneth A. Lambert, Fundamentals of Python: First Programs, CENGAGE Learning, 2nd edition 2012.
4. Charles Dierbach, Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 1st edition 2015.
5. Paul Gries, Jennifer Campbell and Jason Montojo, Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2nd edition 2013. Mapped With: <https://ict.iitk.ac.in/product/python-programming-a-practical>

Online Resources

1. https://onlinecourses.nptel.ac.in/noc21_cs32/preview
2. <https://www.udemy.com/topic/python/>
3. <https://in.coursera.org/courses?query=python>

The bottom of the page contains several handwritten signatures and initials. On the left, there is a signature that appears to be 'V.D.'. In the center, there is a signature that looks like 'R.K.N.'. To the right of that, there are initials 'V.S./h'. Further right, there is a signature that looks like 'A.'. At the bottom right, there is a large, stylized signature that appears to be 'S.P.' followed by a date '18/11/2022'.