NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)

Affiliated to

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY UTTAR PRADESH, LUCKNOW

Evaluation Scheme & Syllabus

For

B.Tech - Second Year-Lateral Entry (B.Sc.)

(Effective from the Session: 2023-24)

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)

B.Tech (CS, CSE(IOT), CSE(CYS),ECE, IT, CSE(AIML), ME, CSE, CSE-R,CSE(DS), CSE(AI), BT)

EVALUATION SCHEME

SEMESTER-III

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BCSE0151Z	Problem Solving using Python	0	0	6				50		100	150	

All the students must clear the above mentioned subjects of the first year (Semester-I) Engineering Program along with the second year (Semester-III) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)

B.Tech (CSBS)

EVALUATION SCHEME

SEMESTER-III

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BCSBS0103Z	Fundamentals of Computer Science	3	0	0	30	20	50		100		150	
2	BCSBS0153Z	Fundamentals of Computer Science Lab	0	0	4				25		25	50	
		GRAND TOTAL										200	

All the students must clear the above mentioned subjects of the first year (Semester-I) Engineering Program along with the second year (Semester-III) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

Abbreviation Used: -

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., PE: Practical End Semester Exam.

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR

(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)

B.Tech (CS, CSE(CYS) , IT, CSE(AIML,CSE, CSE-R,CSE(DS), CSE(AI)

EVALUATION SCHEME

SEMESTER-IV

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BCSE0252Z	Advanced Python	0	0	6				50		100	150	

All the students must clear the above mentioned subjects of the first year (Semester-II) Engineering Program along with the second year (Semester-IV) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR

(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)

B.Tech (BT)

EVALUATION SCHEME

SEMESTER-IV

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BBT0101Z / BBT0102Z	Elementary Mathematics/Remedial Biology	3	1	0	30	20	50		100		150	
		GRAND TOTAL										150	

All the students must clear the above mentioned subjects of the first year (Semester-I) Engineering Program along with the second year (Semester-IV) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

Abbreviation Used: -

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., PE: Practical End Semester Exam.

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR

(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)
B. Tech (ECE, ME, IOT)
EVALUATION SCHEME
SEMESTER-IV

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BCSE0251Z	C Programming	0	0	6				50		100	150	

All the students must clear the above mentioned subjects of the first year (Semester-II) Engineering Program along with the second year (Semester-IV) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

NOIDA INSTITUTE OF ENGG. & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR

(AN AUTONOMOUS INSTITUTE)

Bridge Courses for Lateral Entry Students Admitted Through (B. Sc.)
B.Tech (CSBS)
EVALUATION SCHEME
SEMESTER-IV

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	BCSBS0203Z	Data Structures & Algorithms	3	1	0	30	20	50		100		150	
2	BCSBS0253Z	Data Structures & Algorithms Lab	0	0	4				25		25	50	
		GRAND TOTAL										200	

All the students must clear the above mentioned subjects of the first year (Semester-II) Engineering Program along with the second year (Semester-IV) subjects.

All Bridge Courses (**Compulsory Audit Courses**) a qualifying exam has no credit.
Total and obtained marks are not added in the Grand Total.

Abbreviation Used: -

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., PE: Practical End Semester Exam.

<table>
<tr><td colspan="7"><b>B. Tech.- First Semester
Branch- CSE/CSE-R/CS/IT/M.Tech.(int.)/CSE(IOT)/CSE(DS)/CSE(AI)/CSE((AIML)/CYS/ECE/ECE(VLSI)/ME/BT</b></td></tr>
<tr><td colspan="5"><b>Subject Code-BCSE0151Z</b></td><td colspan="2"><b>L - T - P
0 – 0 – 6</b></td></tr>
<tr><td colspan="5"><b>Subject Name- Problem Solving using Python</b></td><td colspan="2"><b>No. of hours-68</b></td></tr>
<tr><td colspan="7"><b>Course Objective-</b> To provide Basic knowledge of Python programming and to implement programming skills for solving real-world problems.</td></tr>
<tr><td colspan="7"><b>Course Outcome –</b>
<b>CO1 -</b> Understanding basic programming logic.
<b>CO2-</b> Implement python programs using decision control statements.
<b>CO3-</b> Implement user defined functions and modules in python.
<b>CO4-</b> Implement python data structures –lists, tuples, set, dictionaries.
<b>CO5-</b> Apply programming concepts to solve real world problem</td></tr>
<tr><td colspan="7"><b>Course Content</b></td></tr>
<tr><td><b>Unit</b></td><td><b>Module</b></td><td><b>Topics Covered</b></td><td><b>Pedagogy</b></td><td><b>Lecture Required (T=L+P)</b></td><td><b>Aligned Practical/Assignment /Lab</b></td><td><b>CO Mapping</b></td></tr>
<tr><td>Unit 1.</td><td><b>Basics of python programming</b></td><td>Problem Solving, Techniques, Algorithm, Building blocks of algorithms (statements, state, control flow, functions), Notation, Flow chart, Pseudo code, programming language, Categories of programming languages.</td><td>Lecture , Hands-on exercise, Demonstration, practical lab</td><td>6(4+2)</td><td>Implementation of basic Python programs.</td><td>1</td></tr>
</table>

		A Brief History of Python, Applications areas of python, The Programming Cycle for Python, Python IDE, Interacting with Python Programs.		3(1+2)	Installation of IDE and Command Prompt.	1
		Elements of Python: keywords and identifiers, variables, data types and type conversion,		3(1+2)	Demonstrate the use of these in python programs.	1
		operators in python, expressions in python, strings.		3(1+2)	Develop python program to demonstrate use of Operators.	1
Unit 2	**Decision Control Statements**	Conditionals: Conditional statement in Python (if-else statement, its working and execution)	Hands-on exercise, Demonstration, lectures, practical lab	3(1+2)	Develop programs for the use of conditional statements.	2
		Nested-if statement and elif statement in Python, Expression Evaluation & Float Representation.		4(1+3)	Develop programs of different types of statements.	2
		Loops: Purpose and working of loops, while loop, For		7(2+5)	Hands on practice on Loops.	2

		Loop, Nested Loops, Break and Continue, pass statement.				
Unit 3	**Function and Modules**	Introduction of Function, calling a function, Function arguments, built in function, scope rules	Lecture , Hands-on exercise, Demonstration, practical lab	4(1+3)	Learn about how to call or create the functions.	3
		Passing function to a function, recursion, Lambda functions		7(4+3)	Hands-on functions .	
		Modules and Packages: Importing Modules, writing own modules, Standard library modules, dir( ) Function, Packages in Python		4(1+3)	Develop python programs for modules.	
Unit 4	**Basic Data structures in Python**	Strings: Basic operations, Indexing and Slicing of Strings, Comparing strings	Lecture , Hands-on exercise, Demonstration, practical lab	3(1+2)	Implement and play with strings.	4
		Regular expressions. Python Basic Data Structure: Sequence, Unpacking Sequences, Mutable Sequences,		4(1+3)	Demonstration of the regular expression.	

		Lists, Looping in lists, Tuples, Sets, Dictionaries. Map, filter, Reduce, Comprehension		7(3+4)	Implement different methods for these data structures.	
Unit 5	**File and Exception handling**	Files and Directories: Introduction to File Handling in Python, Reading and Writing files, Additional file methods, Working with Directories.	Lecture , Hands-on exercise, Demonstration, practical lab	4(1+3)	Learn Python file handling methods and python file operations	5
		Exception Handling, Errors, Run Time Errors, Handling IO Exception, Try-except statement, Raise		6(2+4)	Learn about Python exception handling methods	5

References-

Text Books:

1. Magnus Lie Hetland, "Beginning Python-From Novice to Professional"—Third Edition, Apress
2. Python Programming using Problem solving approach by ReemaThareja OXFORD Higher education
3. Kenneth A. Lambert, —Fundamentals of Python: First Programs, CENGAGE Learning, 2012.

Reference Books:

1. John V Guttag, —Introduction to Computation and Programming Using Python‟, Revised and expanded Edition, MIT Press , 2013
2. Charles Dierbach, —Introduction to Computer Science using Python: A Computational Problem Solving Focus, Wiley India Edition, 2013.
3. Allen B. Downey, "Think Python: How to Think Like a Computer

Links:

UNIT 1: https://nptel.ac.in/courses/106/106/106106182/

UNIT 2: https://nptel.ac.in/courses/106/106/106106212/
https://www.youtube.com/watch?v=PqFKRqpHrjw

UNIT 3: https://nptel.ac.in/courses/106/106/106106145/
https://www.youtube.com/watch?v=m9n2f9lhtrw
https://www.youtube.com/watch?v=oSPMmeaiQ68
UNIT 4: https://nptel.ac.in/courses/106/106/106106145/
https://www.youtube.com/watch?v=ixEeeNjjOJ0&t=4s
UNIT 5: https://nptel.ac.in/courses/106/106/106106145/
https://www.youtube.com/watch?v=NMTEjQ8-AJM

LAB

Total No. of Practicals : 228

List of Practicals

Lab No.	**Unit**	**Topic**	**Program Logic Building**	**CO Mapping**
1.1	1	Basic Python(Syntax, Variable, Type Conversion)	Python Program to Print Statement	CO1
1.2	1	Basic Python(Syntax, Variable, Type Conversion)	Swap two variables without using a temporary variable.	CO1
1.3	1	Basic Python(Syntax, Variable, Type Conversion)	Check if a given number is even or odd.	CO1
1.4	1	Basic Python(Syntax, Variable, Type Conversion)	Find the largest of three numbers.	CO1
1.5	1	Basic Python(Syntax, Variable, Type Conversion)	Convert a string to an integer.	CO1
1.6	1	Basic Python(Syntax, Variable, Type Conversion)	Convert an integer to a string.	CO1
1.7	1	Basic Python(Syntax, Variable, Type	Convert a string to a floating-point number.	CO1

		Conversion)		
1.8	1	Basic Python(Syntax, Variable, Type Conversion)	Convert a floating-point number to an integer.	CO1
1.9	1	Basic Python(Syntax, Variable, Type Conversion)	WAP to demonstrate implicit and explicit type conversion.	CO1
1.10	1	Basic Python(Syntax, Variable, Type Conversion)	Convert Employee Count to Binary	CO1
1.11	1	Basic Python(Syntax, Variable, Type Conversion)	Convert Revenue to Currency Format	CO1
1.12	1	Operators	Write a program to Calculate Sum of 5 Subjects and Find Percentage (Max Mark in each subject is 100).	CO1
1.13	1	Operators	Write a program to find gross salary.	CO1
1.14	1	Operators	Write a program to Calculate Area of Rectangle, Square.	CO1
1.15	1	Operators	Write a program to Calculate Area of Scalene Triangle and Right-angle Triangle.	CO1
1.16	1	Operator	Write a program to find the perimeter of a circle, rectangle and triangle.	CO1
1.17	1	Operator	Write a program to Compute Simple Interest.	CO1
1.18	1	Operator	Write a program to Convert Fahrenheit temperature in to Celsius.	CO1
1.19	1	Operator	Write a program to Find the Gravitational Force Acting Between Two Objects.	CO1
1.20	1	Operator	Write a program to swap the values of two variables with and without using third variable.	CO1
1.21	1	Operator	Write a program to perform arithmetic operations on a = 8, b = 3.	CO1
1.22	1	Operator	Write a program to apply relational operations on a=8, b=3.	CO1

1.23	1	Operator	Write a program to apply assignment operations on a=8, b=3.	CO1
1.24	1	Operator	Write a program to apply logical operations on a=8, b=3.	CO1
1.25	1	Operator	Write a program to apply bitwise operations on a=8, b=3.	CO1
1.26	1	Operator	Write a program to apply identity operators.	CO1
1.27	1	Operator	Write a program to Swap the Contents of two Numbers using Bitwise XOR Operation	CO1
1.28	1	Operator	WAP to find the absolute value of the given number.	CO1
1.29	1	Operator	Write a program to Add two Complex Numbers.	CO1
1.30	1	Operator	Write a Program to find roots of a quadratic expression.	CO1
1.31	1	Arithmetic Operator	Program to perform basic arithmetic operations (addition, subtraction, multiplication, division) on two numbers.	CO1
1.32	1	Arithmetic Operator	Program to calculate the area of a rectangle using the multiplication operator.	CO1
1.33	1	Arithmetic Operator	Program to calculate the average of a list of numbers using the division operator.	CO1
1.34	1	Comparison Operator	Program to compare two numbers and determine if they are equal.	CO1
1.35	1	Comparison Operator	Program to compare two numbers and determine whether they are greater than or less than .	CO1
1.36	1	Comparison Operator	Program to check if a given string is equal to a specific value.	CO1
1.37	1	Logical Operator	Write a program to apply Logical AND operator on two operands.	CO1
1.38	1	Logical Operator	Write a program to apply Logical OR operator on two operands.	CO1

1.39	1	Logical Operator	Write a program to apply Logical NOT operator on an operand.	CO1
1.40	1	Assignment operator	Program to increment or decrement a variable using assignment operators.	CO1
1.41	1	Assignment operator	Program to calculate compound interest using compound assignment operators.	CO1
1.42	1	Bitwise Operator	Program to perform bitwise AND, OR, XOR, left shift, and right shift operations.	CO1
1.43	1	Bitwise Operator	Program to check if a given number is odd or even using bitwise operators.	CO1
2.1	2	Conditional Statements	Write a program to Accept two Integers and Check if they are Equal.	CO 2
2.2	2	Conditional Statements	Write a program to Check if a given Integer is Positive or Negative and Odd or Even.	CO 2
2.3	2	Conditional Statements	Write a program to Check if a given Integer is Divisible by 7 or not.	CO 2
2.4	2	Conditional Statements	Write a program to find the greatest of three numbers using else if ladder.	CO 2
2.5	2	Conditional Statements	Write a program to find the greatest of three numbers using Nested if.	CO 2
2.6	2	Conditional Statements	Write a program to convert an Upper-case character into lower case and vice-versa.	CO 2
2.7	2	Conditional Statements	Write a program to check weather an entered year is leap year or not.	CO 2
2.8	2	Conditional Statements	Write a Program to check whether an alphabet entered by the user is a vowel or a constant.	CO 2
2.9	2	Conditional Statements	Write a program to print day according to the day number entered by the user.	CO 2
2.10	2	Conditional Statements	Write a program to print color name, if user enters the first letter of the color name.	CO 2
2.11	2	Conditional Statements	Write a program to Simulate Arithmetic	CO 2

			Calculator.	
2.12	2	Conditional Statements	Write a menu driven program for calculating area of different geometrical figures such as circle, square, rectangle, and triangle.	CO 2
2.13	2	Conditional Statements	WAP that accepts the marks of 5 subjects and finds the percentage marks obtained by the student. It also prints grades according to the following criteria: Between 90-100% Print 'A', 80-90% Print 'B', 60-80% Print 'C', 50-60% Print 'D', 40-50% Print 'E', Below 40% Print 'F’.	CO 2
2.14	2	Conditional Statements	WAP to enter a character and then determine whether it is a vowel, consonants, or a digit.	CO 2
2.15	2	Loops	Write a program to display all even numbers from 1 to 20	CO 2
2.16	2	Loops	Write a program to print all the Numbers Divisible by 7 from 1 to 100.	CO 2
2.17	2	Loops	Write a program to print table of any number.	CO 2
2.18	2	Loops	Write a program to Find the Sum of first 50 Natural Numbers using for Loop.	CO 2
2.19	2	Loops	Write a program to calculate factorial of a given number using for loop and also using while loop.	CO 2
2.20	2	Loops	Write a program to count the sum of digits in the entered number.	CO 2
2.21	2	Loops	Write a program to find the reverse of a given number.	CO 2
2.22	2	Loops	Write a program to Check whether a given Number is Perfect Number.	CO 2
2.23	2	Loops	Write a program to Print Armstrong Number from 1 to 1000.	CO 2
2.24	2	Loops	Write a program to Compute the Value of X^n.	CO 2
2.25	2	Loops	Write a program to Calculate the value of ^nC_r.	CO 2
2.26	2	Loops	Write a program to generate the Fibonacci Series.	CO 2
2.27	2	Loops	Write a program to check whether a given	CO 2

			Number is Palindrome or Not.	
2.28	2	Loops	Write a program to Check whether a given Number is an Armstrong Number.	CO 2
2.29	2	Loops	Write a program to print all prime numbers from 1-500.	CO 2
2.30	2	Loops	Write a program to find the Sum of all prime numbers from 1-1000.	CO 2
2.31	2	Loops	Write a program to display the following pattern: *	CO 2
2.32	2	Loops	Write a program to display the following pattern: * * * * * * * * * * * * * * *	CO 2
2.33	2	Loops	Write a program to display the following pattern: 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	CO 2
2.34	2	Loops	Write a program to display the following pattern: A B B C C C D D D D E E E E E	CO 2

2.35	2	Loops	Write a program to display the following pattern: * * * * * * * * * * * * * * *	CO 2
2.36	2	Loops	Write a program to display the following pattern: 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	CO 2
2.37	2	Loops	Write a program to display the following pattern: * * * * * * * * * * * * * * * *	CO 2
2.38	2	Loops	Write a program to display the following pattern: *	CO 2
2.39	2	Loops	Write a program to display the following pattern (Pascal Triangle): 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1	CO 2
2.40	2	Loops	Write a program to display the following pattern:	CO 2

			1 2 3 4 5 6 7 8 9 10	
2.41	2	Loops	Write a program to display the following pattern: A B C D E F G F E D C B A A B C D E F F E D C B A A B C D E E D C B A A B C D D C B A A B C C B A A B B A A A	CO 2
2.42	2	Loops	Write a program to display the following pattern: *	CO 2
2.43	2	Loops	Write a program to display the following pattern: 0 0 01 10 010 010 0101 1010 0101001010	CO 2
2.44	2	Loops	Write a program to display the following pattern:	CO 2

			A B C D E F G H I J K L M N O	
2.45	2	Loops	Write a program to display the following pattern: A BAB CBABC DCBABCD EDCBABCDE	CO 2
2.46	2	Loops	Write a program to Find the Sum of A.P Series.	CO 2
2.47	2	Loops	Write a program to Find the Sum of G.P Series.	CO 2
2.48	2	Loops	Write a program to Find the Sum of H.P Series.	CO 2
2.49	2	Loops	Write a program to print the following sequence of integers. 1, 2, 4, 8, 16, 32	CO 2
2.50	2	Loops	Write a program to find the Sum of following Series: (1*1) + (2*2) + (3*3) + (4*4) + (5*5) + ... + (n*n)	CO 2
2.51	2	Loops	Write a program to find the Sum of following Series: (1^1) + (2^2) + (3^3) + (4^4) + (5^5) + ... + (n^n)	CO 2
2.52	2	Loops	Write a program to find the Sum of following Series: (1!/1) + (2!/2) + (3!/3) + (4!/4) + (5!/5) + ... + (n!/n)	CO 2
2.53	2	Loops	Write a program to print the following Series: 1, 2, 3, 6, 9, 18, 27, 54, ... upto n terms	CO 2
2.54	2	Loops	Write a program to print the following Series: 2, 15, 41, 80, 132, 197, 275, 366, 470, 587	CO 2

2.55	2	Loops	Write a program to print the following Series:1, 3, 4, 8, 15, 27, 50, 92, 169, 311	CO 2
2.56	2	Loops	Write a program to Convert the given Binary Number into Decimal.	CO 2
2.57	2	Loops	Write a program to Convert Binary to Hexadecimal.	CO 2
2.58	2	Loops	Write a program to find out L.C.M. of two numbers.	CO 2
2.59	2	Loops	Write a program to find out H.C.F. of two numbers.	CO 2
2.60	2	Loops	Python Program to Accept Three Digits and Print all Possible Combinations from the Digits.	CO 2
2.61	2	Loops	Python Program to Print Odd Numbers within a Given Range.	CO 2
2.62	2	Loops	Python Program to Find the Smallest Divisor of an Integer.	CO 2
2.63	2	Loops	Python Program to Count the Number of Digits in a Number	CO 2
2.64	2	Loops	Python program to find GCD between two given integer numbers.	CO 2
3.1	3	Functions	Write a Python function to find the Max of three numbers.	CO3
3.2	3	Functions	Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7) Expected Output : 20	CO3
3.3	3	Functions	Write a Python program to reverse a string. Sample String : "1234abcd" Expected Output : "dcba4321"	CO3
3.4	3	Functions	Write a Python function to check whether a number falls in a given range.	CO3
3.5	3	Functions	Write a Python function that accepts a string and calculate the number of upper-case letters and	CO3

			lower-case letters. Sample String: 'The quick Brow Fox' Expected Output : No. of Upper case characters : 3 No. of Lower case Characters : 1	
3.6	3	Functions	Write a Python function that takes a number as a parameter and check the number is prime or not.	CO3
3.7	3	Functions	Write a Python function that checks whether a passed string is palindrome or not.	CO3
3.8	3	Functions	Write a Python function that prints out the first n rows of Pascal's triangle.	CO3
3.9	3	Functions	Write a Python function that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically. *Sample Items:* green-red-yellow-black-white Expected Result: black-green-red-white-yellow	CO3
3.10	3	Functions	Python function to convert height (in feet and inches) to centimeters	CO3
3.11	3	Functions	Python function to Convert Celsius to Fahrenheit.	CO3
3.12	3	Functions	Implement a function to check if two strings are anagrams of each other.	CO3
3.13	3	Functions	Python function to display all the Armstrong number from 1 to n.	CO3
3.14	3	Recursion	Write a program using recursion to compute factorial of a given number.	CO3
3.15	3	Recursion	Write a program to print Fibonacci Series using recursion.	CO3
3.16	3	Recursion	Write a program to calculate sum of numbers 1 to N using recursion.	CO3
3.17	3	Recursion	Write a program to Find Sum of Digits of the Number using Recursive Function.	CO3
3.18	3	Recursion	Write a program to print Tower of Hanoi using	CO3

			recursion.	
3.19	3	Recursion	Python Program to Determine How Many Times a Given Letter Occurs in a String Recursively	CO3
3.20	3	Recursion	Python Program to Find the Binary Equivalent of a Number Recursively	CO3
3.21	3	Recursion	Python Program to Find the GCD of Two Numbers Using Recursion	CO3
3.22	3	Recursion	Python Program to Find the Power of a Number Using Recursion	CO3
3.23	3	Recursion	WAP to compute the sum of all the elements of the list using reduce() function.	CO3
3.24	3	Modules and Pacakges	A) Write a program to create a module and import the module in another python program.	CO3
3.25	3	Modules and Pacakges	Write a program program to import all objects from a modules, specific objects from module and provide custom import name to the imported object from the module.	CO3
3.26	3	Modules and Pacakges	Create a python package having atleast two modules in it.	CO3
3.27	3	Modules and Pacakges	Create a python package having atleast one subpackage in it.	CO3
4.1	4	String	Python program to check whether the string is Symmetrical or Palindrome	CO 4
4.2	4	String	Ways to remove i’th character from string in Python	CO 4
4.3	4	String	Python program to Check if a Substring is Present in a Given String	CO 4
4.4	4	String	Find length of a string in python (4 ways)	CO 4
4.5	4	String	Python program to print even length words in a	CO 4

			string	
4.6	4	String	Python program to accept the strings which contains all vowels	CO 4
4.7	4	String	Remove all duplicates from a given string in Python	CO 4
4.8	4	String	Python program to Maximum frequency character in String	CO 4
4.9	4	String	Python Program to Replace all Occurrences of ‘a’ with $ in a String	CO 4
4.10	4	String	Python Program to Form a New String where the First Character and the Last Character have been Exchanged	CO 4
4.11	4	String	Python Program to Count the Number of Vowels in a String	CO 4
4.12	4	String	Python Program to Take in a String and Replace Every Blank Space with Hyphen	CO 4
4.13	4	String	Python Program to Calculate the Length of a String Without Using a Library Function	CO 4
4.14	4	String	Python Program to Remove the Characters of Odd Index Values in a String	CO 4
4.15	4	String	Python Program to Calculate the Number of Words and the Number of Characters Present in a String	CO 4
4.16	4	String	Python Program to Take in Two Strings and Display the Larger String without Using Built-in Functions	CO 4
4.17	4	String	Python Program to Check if a String is a Pangram or Not (A pangram is a sentence that uses all 26 letters of the English alphabet at least once. like” The quick brown fox jumps over the lazy dog”)	CO 4
4.18	4	String	Python Program to Accept a Hyphen Separated Sequence of Words as Input and Print the Words in a Hyphen-Separated Sequence after Sorting	CO 4

			them Alphabetically	
4.19	4	String	Python Program to Form a New String Made of the First 2 and Last 2 characters From a Given String	CO 4
4.20	4	String	Python Program to Count the Occurrences of Each character in a Given String Sentence	CO 4
4.21	4	String	Python Program to Check if a Substring is Present in a Given String	CO 4
4.22	4	String	Python Program to Find the Most Repeated Word in a String.	CO 4
4.23	4	Regular Expression	Write a python program to check the validity of a password given by the user. The password should satisy the following criteria: i) Contain atleast 1 letter between a and z. ii) Contain atleast 1 number between 0 and 9. iii) Contain atleast 1 letter between A and Z. iv) Contain atleast 1 character from $,#,@. v) Maximum length of password 6. vi) Maximum length of password:12.	CO 4
4.24	4	Regular Expression	Write a python program to validate mobile number.	CO 4
4.25	4	Regular Expression	Given an input file which contains a list of names and phone numbers separated by spaces in the following: i) Phone number contains a 3- or 2-digit area code and a hyphen followed by an 8-digit number. **ii)** Find all names having phone number with a 3digit area code using regular expression.	CO 4
4.26	4	List	Program to interchange first and last elements in a list	CO 4

4.27	4	List	WAP to find min, max and average of elements of a list having numeric data	CO 4
4.28	4	List	Program to check if element exists in list	CO 4
4.29	4	List	Program for Reversing a List	CO 4
4.30	4	List	Program to Multiply all numbers in the list	CO 4
4.31	4	List	Program to find smallest and largest number in a list	CO 4
4.32	4	List	Program to find second largest number in a list	CO 4
4.33	4	List	Program to print all even numbers in a range	CO 4
4.34	4	List	Program to print all negative numbers in a range	CO 4
4.35	4	List	Program to Remove multiple elements from a list in Python	CO 4
4.36	4	List	Program to Cloning or Copying a list	CO 4
4.37	4	List	Program to Count occurrences of an element in a list	CO 4
4.38	4	List	Program to find Cumulative sum of a list	CO 4
4.39	4	List	Program to Break a list into chunks of size N in Python	CO 4
4.40	4	List	Python Program to transpose of Matrix.	CO 4
4.41	4	List	Python Program to Add Two Matrices.	CO 4
4.42	4	List	Python Program to Multiply Two Matrices.	CO 4
4.43	4	List	Program to get K^{th} Column of Matrix	CO 4
4.44	4	List	WAP to print all even numbers of a list using list comprehension.	CO 4
4.45	4	List	WAP that prompts user to enter an alphabet and then print all the words that starts with that alphabet from the list of words.	CO 4
4.46	4	List	WAP to transpose a given matrix using list comprehension.	CO 4
4.47	4	List	Print All the characters of a string using list Comprehension	CO 4
4.48	4	List	Write a program to calculate square of numbers upto n using list comprehension.	CO 4

4.49	4	Tuple	Python program to Find the size of a Tuple	CO 4
4.50	4	Tuple	Python – Maximum and Minimum K^{th} elements in Tuple	CO 4
4.51	4	Tuple	Create a list of tuples from given list having number and its cube in each tuple	CO 4
4.52	4	Tuple	Python – Flatten tuple of List to tuple	CO 4
4.53	4	Set	Python Program to Count the Number of Vowels Present in a String using Sets	CO 4
4.54	4	Set	Python Program to Check Common Letters in Two Input Strings	CO 4
4.55	4	Set	Python Program that Displays which Letters are in the First String but not in the Second	CO 4
4.56	4	Set	Python Program that Displays which Letters are Present in Both the Strings	CO 4
4.57	4	Set	Python Program that Displays which Letters are in the Two Strings but not in Both	CO 4
4.58	4	Dictionary	Python Program to Add a Key-Value Pair to the Dictionary	CO 4
4.59	4	Dictionary	Python Program to Concatenate Two Dictionaries into One.	CO 4
4.60	4	Dictionary	Python Program to Check if a Given Key Exists in a Dictionary or Not	CO 4
4.61	4	Dictionary	Python Program to Generate a Dictionary that Contains Numbers (between 1 and n) in the Form (x,x*x).	CO 4
4.62	4	Dictionary	Python program to create an instance of an Ordered dict using a given dictionary. Sort the dictionary during the creation and print the members of the dictionary in reverse order.	CO 4
4.63	4	Dictionary	Python Program to Sum All the Items in a Dictionary	CO 4
4.64	4	Dictionary	WAP to create dictionary which has characters of given string as keys and frequency of characters as	CO 4

			values.	
4.65	4	Dictionary	Python Program to Multiply All the Items in a Dictionary	CO 4
4.66	4	Dictionary	Python Program to Remove the Given Key from a Dictionary	CO 4
4.67	4	Dictionary	Python Program to Form a Dictionary from an Object of a Class	CO 4
4.68	4	Dictionary	Python Program to Map Two Lists into a Dictionary	CO 4
4.69	4	Comprehension	Write a program Filtering even numbers from a list using tuple comprehension	CO 4
4.70	4	Comprehension	Creating a list of tuples from two lists using comprehension function	CO 4
4.71	4	Comprehension	Extracting the first character from each word in a list of strings	CO 4
4.72	4	Comprehension	Swapping keys and values in a dictionary	CO 4
4.73	4	Comprehension	Filtering even numbers from a dictionary:	CO 4
4.74	4	Comprehension	Write a Program to calculate square of number using dictonary comprehension	CO 4
5.1	5	File handling and Exceptional Handling	Python program to read file word by word	CO 5
5.2	5	File handling and Exceptional Handling	Python program to read character by character from a file	CO 5
5.3	5	File handling and Exceptional Handling	Python – Get number of characters, words, spaces and lines in a file	CO 5
5.4	5	File handling and Exceptional Handling	Program to Find ‘n’ Character Words in a Text File	CO 5
5.5	5	File handling and Exceptional Handling	Python Program to obtain the line number in which given word is present	CO 5
5.6	5	File handling and Exceptional Handling	Count number of lines in a text file in Python	CO 5
5.7	5	File handling and Exceptional Handling	Python Program to remove lines starting with any prefix	CO 5

5.8	5	File handling and Exceptional Handling	Python Program to Eliminate repeated lines from a file	CO 5
5.9	5	File handling and Exceptional Handling	Python Program to read List of Dictionaries from File	CO 5
5.10	5	File handling and Exceptional Handling	Python – Append content of one text file to another	CO 5
5.11	5	File handling and Exceptional Handling	Python program to copy odd lines of one file to other	CO 5
5.12	5	File handling and Exceptional Handling	Python Program to merge two files into a third file	CO 5
5.13	5	File handling and Exceptional Handling	Python program to Reverse a single line of a text file	CO 5
5.14	5	File handling and Exceptional Handling	Python program to reverse the content of a file and store it in another file	CO 5
5.15	5	File handling and Exceptional Handling	Python Program to handle divide by zero exception.	CO 5
5.16	5	File handling and Exceptional Handling	WAP to handle multiple exception.	CO 5
5.17	5	File handling and Exceptional Handling	Python program to combine each line from first file with the corresponding line in second file.	CO 5
5.18	5	File handling and Exceptional Handling	Write a program to copy the contents of one file to another.	CO 5
5.19	5	File handling and Exceptional Handling	Write a program to print First 5 line in a file	CO 5
5.20	5	File handling and Exceptional Handling	a) Write a program to catch the following exception: i) Value error ii) Index error iii) Name error iv) Type error v) Divide zero error b) Write a program to create user defined exceptions.	CO 5

			c) Write a program to understand the use of else and finally block with try block. **d)** Write a python program that uses raise and exception class to throw an exception.	

B. Tech.-Second Semester **Branch- CS/ CSE/CSE (R)/ IT/CSE(DS)/CSE(IOT)/CSE(AIML)/CSE(AI)/CYS/ ECE/ECE(VLSI)/ ME/M. Tech (Integrated)/ BT**	
Subject Code-BCSE0252Z	**L - T - P** **0 –0 - 6**
Subject Name- Advanced Python	**No. of hours- 78 hours**
Course Objective- To become familiar with Python's Object-Oriented Concepts, functional programming And create GUI application and to gain the knowledge of Python libraries.	
Course Outcome – **CO1 -** Implement classes and create instances in python **CO2-** Implement GUI based Python application **CO3-** Use Python libraries for data handling. **CO4-** Analyze data using visualization libraries. **CO5-** Analyze web scraping application for real world data	
Course Content	

Unit	**Module**	**Topics Covered**	**Pedagogy**	**Lecture Required (T=L+P)**	**Aligned Practical/Assignment/Lab**	**CO Mapping**
Unit 1	**Classes and Objects**	Introduction: Python Classes and objects, User-Defined Classes, Class Variables and Instance Variables	Lecture , Hands-on exercise, Demonstration, practical lab	4(3+1)	Learn to create python classes and objects.	1

		Instance methods, Class method, static methods,		4(2+2)	Perform different types of class methods.	1
		constructor in python, parametrized constructor, Magic Methods in python,		3(3+2)	Create a constructor to initialize an object in Python, Different types of constructors, Constructor overloading and chaining	1
		Object as an argument, Instances as Return Values, namespaces,		2(1+1)	Implementation of Object as an argument, Instances and namespace	1
		Introduction to inheritance and polymorphism, **Abstract Class, Introduction to Abstraction and Encapsulation**		8(3+5)	Implementing inheritance and types of polymorphism.	1
Unit 2	**Functional and GUI Programming**	**Functional Programming**: Immutability, Closures and Decorators, generators	Hands-on exercise, Demonstration, lectures, practical lab	6(2+4)	Implementation of Decorators and generators	2

		Co-routines, iterators, Declarative programming		3(2+1)	Implement the functions of iterators and co routines	2
		GUI Programming: Intro to GUI Programming , Settling widgets in the window's interior, Numeric Widgets,		3(0+3)	Demonstration of GUI interface.	2
		Boolean Widgets, Selection Widgets, String Widgets, Date Picker, Color Picker, Container Widgets,		2(0+2)	Implement different types of GUI widgets.	2
		Creating a GUI Application, Tkinter, button, canvas		2(0+2)	Create GUI application using Tkinter and components.	2
Unit 3	**Libraries for Data Handling**	**NumPy:** Basic Operation, Indexing, slicing and Iterating	Lecture , Hands-on exercise, Demonstration, practical lab	3(1+2)	Demonstration on numpy, and mathematical operations on numpy.	3
		Multidimensional arrays, NumPy Data types, Reading and writing data		3(1+2)	Implementation of Multi-dimensional array.	

		on Files				
		SciPy: Introduction to SciPy, Create function, modules of SciPy.		3(1+2)	Learn to demonstrate the SciPy libraries.	
		Pandas : Series and Data Frames, Grouping, aggregation, Merge Data Frames,		3(1+2)	Learn to demonstrate the use of pandas, data frames..	
		Generate summary tables, Group data into logical pieces, Manipulation of data		3(1+2)	Creating tables and groups.	
Unit 4	**Libraries in Data Visualization**	**Matplotlib:** Scatter plot, Bar charts, histogram, Stack charts	Lecture , Hands-on exercise, Demonstration, practical lab	3(1+2)	Learn to demonstrate the different visualization methods.	4
		Legend title Style, Figures and subplots,		1(0.5+0.5)	Implementation on charts and figures.	4
		Plotting function in pandas, Labelling and arranging figures, Save plots.		3(1+2)	Implementation on plots and figures.	4

		Seaborn: style function, color palettes, heatmaps ,distribution plots, category plot, regression plot		3(1+2)	Implementation of seaborn library	4
		Plotly : Lineplots , Areaplots, Scatterplots, Bubbleplots , Stacked bar charts,		2(1+1)	Implementation of different types of plots.	4
		Grouped bar charts, Pie charts, Tables, Dashboards		2(1+1)	Implementation of charts.	4
Unit 5	**Web Scraping with Python**	Web Scraping: Introduction, Web Crawling v/s Web Scraping, Uses of Web Scraping, Components of a Web Scraper, working of a Web Scraper, Crawl, Parse and Transform Store the Data	Lecture , Hands-on exercise, Demonstration, practical lab	3(1+2)	Learn to scrap the data.	5

		Beautiful Soup: Introduction to Beautiful Soup library, Accessing Tags, Navigable Strings, Navigating and searching with Beautiful Soup, Web Scraping		3(1+2)	Demonstration of web scrapping using Beautiful Soup.	5
		Example: Scraping Flipkart Website		4(1+3)	Learn to scrapping of Flipkart website.	5
		Introduction to Github		2(1+1)	Implementation of Projects on Github.	5

References-

Text Books:

1. Magnus Lie Hetland, "Beginning Python-From Novice to Professional"—Third Edition, Apress
2. Peter Morgan, Data Analysis from Scratch with Python, AI Sciences
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd edition, Updated for Python 3, Shroff/O‘Reilly Publishers, 2016
4. Miguel Grinberg, Developing Web applications with python, OREILLY

Reference Books:

1. Dusty Phillips, Python 3 Object-oriented Programming - Second Edition, O’Reilly
2. Burkhard Meier, Python GUI Programming Cookbook - Third ,Packt
3. DOUG HELLMANN, THE PYTHON 3 STANDARD LIBRARY BY EXAMPLE, :Pyth 3 Stan Libr Exam _2 (Developer's Library) 1st Edition, Kindle Edition
4. Kenneth A. Lambert, —Fundamentals of Python: First Programsۅ, CENGAGE Learning, 2012.

Links:

Unit 1	https://nptel.ac.in/courses/106/106/106106145/
Unit 2	https://www.python-course.eu/python3_inheritance.php
Unit 3	https://realpython.com/courses/functional-programming-python/
Unit 4	https://realpython.com/python-gui-tkinter/
Unit 5	https://nptel.ac.in/courses/106/107/106107220/ https://nptel.ac.in/courses/106/106/106106212/

B.TECH FIRST YEAR

		L	T	P	Credit
Course Code	**BCSBS0103Z**	**L**	**T**	**P**	**Credit**
Course Title	**Fundamentals of Computer Science**	**3**	**0**	**0**	**0**

Course objective:

The course covers various operations, conditional statements and looping constructs in C. The course aims to solve complex problems using functions and arrays in C.

Pre-requisites:Basic Knowledge of Computer

Course Contents / Syllabus

UNIT-I	**General problem Solving concepts**	**5 hours**

Algorithm, and Flowchart for problem solving with Sequential Logic Structure, Decisions and Loops.
Programming using C: applications of C programming, Structure of C program, Overview of compilation and execution process in an IDE, transition from algorithm to program, Syntax, logical errors and Run time errors, object and executable code

UNIT-II	**Imperative languages&Operators**	**7 hours**

Introduction to imperative language; syntax and constructs of a specific language (ANSI C)
Types Operator and Expressions with discussion of variable naming and Hungarian Notation: Variable Names, Data Type and Sizes (Little Endian Big Endian), Constants, Declarations, Arithmetic Operators, Relational Operators, Logical Operators, Type Conversion, Increment Decrement Operators, Bitwise Operators, Assignment Operators and Expressions, Precedence and Order of Evaluation, proper variable naming and Hungarian Notation.

UNIT-III	**Control Flow**	**6 hours**

Control Flow with discussion on structured and unstructured programming: Statements and Blocks, If-Else-If, Switch, Loops – while, do, for, break and continue, goto labels, structured and un- structured programming.

UNIT-IV	**Functions and Program Structure**	**8 hours**

Functions and Program Structure with discussion on standard library: Basics of functions, parameter passing and returning type, C main return as integer, External, Auto, Local, Static, Register Variables, Scope Rules, Block structure, Initialization, Recursion, Pre-processor, Standard Library Functions and return types.

UNIT-V	**Pointers and Arrays**	**8 hours**

Pointers and address, Pointers and Function Arguments, Pointers and Arrays, Address Arithmetic, character Pointers and Functions, Pointer Arrays, Pointer to Pointer, Multi-dimensional array and Row/column major formats, Initialization of Pointer Arrays, Command line arguments, Pointer to functions, complicated declarations and how they are evaluated.
Structures: Basic Structures, Structures and Functions, Array of structures, Pointer of structures, Self-referral structures, Table look up, typedef, unions, Bit-fields

UNIT-VI	**Input and Output:**	**6 Hours**

Standard I/O, Formatted Output – printf, Formated Input – scanf, Variable length argument list, file access including FILE structure, fopen, stdin, stdout and stderr, Error Handling including exit, perror and error.h, Line I/O, related miscellaneous functions.
Unix system Interface: File Descriptor, Low level I/O – read and write, open, create, close and unlink, Random access – seek, Discussions on Listing Directory, Storage allocator.
Programming Method: Debugging, Macro, User Defined Header, User Defined Library Function, makefile utility

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

CO 1	Acquire a broad perspective about the uses of computers in engineering industry.	K2
CO 2	Understand the concept of computers, algorithm and algorithmic thinking.	K2

CO 3	Apply conditional statements and looping constructs.	K3
CO 4	Implement array and perform operations on it.	K3
CO 5	Understand the more advanced features of the C language	K2

Text Books

1. B. W. Kernighan and D. M. Ritchi, The C Programming Language, 1988, 2nd Edition, PHI.
2. B. Gottfried, Programming in C, Schaum Outline Series, 1996, 2nd Edition, McGraw Hill Companies Inc.

Reference Books

1. Herbert Schildt, C: The Complete Reference, 2000, 4th edition, McGraw Hill.
2. YashavantKanetkar, Let Us C, 2017, 15th edition, BPB Publications.

Subject Code- BCSBS0153Z	**L T P** **0 0 4**
Subject Name- Fundamentals of Computer Science Lab	**No. of Hours:**

Course Outcome-

CO1- Read, understand and trace the execution of programs written in C language.

CO2- Write the C code for a given algorithm.

CO3- Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor.

CO4- Write programs that perform operations using derived data types.

CO5- Implement String Handling.

Total No. of Practicals

List of Practicals

Lab No.	Unit	Topic	Program Logic Building	CO Mapping
1	I	1. Algorithm and flowcharts of small problems like GCD	1. Algorithm to find the sum of two numbers 2. Algorithm to find the smallest of two numbers 3. Algorithm to print multiplication Table of a number 4. Algorithm to find GCD of numbers.	1
2	II & III	Structured code writing with: i. Small but tricky codes	1. WAP in C to implement the use of all arithmetic operators. 2. WAP in C to find the Area and Circumference of a circle, where radius r is input by user. 3. WAP in C to swap the values of 2 variables without using third variable. 4. WAP in C to print the digit at ones place of a number. 5. WAP in C to calculate the total amount of money in the piggybank, given the coins of Rs 10, Rs 5, Rs 2 and Rs 1.	1

			6. WAP in C to Enter the marks of 5 subjects (i.e. Physics, Chemistry, Maths, Hindi & English) of a student & display the Total_Marks and Percentage and grade achieved. Grading Critria: percentage >= 90 then Grade A percentage >=80 and <90 then Grade B percentage >=70 and <80 then Grade C percentage >=60 and <70 then Grade D else Grade E 7. WAP in C to check whether a year is Leap Year or not. 8. WAP in C to print day of week name using switch case. 9. WAP in C to print total number of days in a month using switch case. 10. WAP in C to calculate the sum of first 10 numbers. 11. WAP in C to print multiplication table of any number. 12. WAP in C to display a Fibonacci series. 13. WAP in C to find sum of digits of a number. 14. WAP in C to reverse a number. 15. Pattern Printing programs.	
3	IV	ii. Proper parameter passing	1. WAP in C to demonstrate call by value and call by reference.	1
4	V	iii. Command line Arguments	1. WAP in C to demonstrate command line arguments.	1
5	IV	iv. Variable parameter	1. WAP in C to add two number using user defined function add(). 2. WAP in C to find the largest of three number using user defined function largest() . 3. WAP in C to calculate the factorial of a number using recursion. Program to calculate the exponent using recursion.	2
6	V	v. Pointer to functions	1. WAP in C that uses pointer to point address of a function. 2. WAP in C that passes function as a parameter using pointer.	2
7	IV	vi. User defined header	1. WAP in C to create a user defined header file and use it in to some other programs.	3
8	VI	vii. Make file utility	1. WAP in C to implement all make file utility commands.	3
9	IV	viii. Multi file program and user defined libraries	1. Program to demonstrate how to use multiple c files in one program.	4

10	V	ix. Interesting substring matching / searching programs	1. C Program to Implement Knuth-Morris-Pratt Algorithm for Pattern Searching. 2. C Program to Implement KMP Pattern Searching Algorithm. 3. C Program to Implement Rabin-Karp Method for Pattern Searching.	4
11	II	x. Parsing related assignments	1. Create a program that takes a user input string in the form of "operand1 operator operand2" (e.g., "5 + 3" or "10 * 2") and parses it to perform the corresponding mathematical operation. Display the result to the user.	4

B.TECH FIRST YEAR			
Course Code	BCSE0252Z	**L T P**	**Credit**
Course Title	**Fundamentals of Computer Science Lab**	**0 0 4**	**0**

Suggested List of Experiments	**CO**
1. Algorithm and flowcharts of small problems like GCD	1
2. Structured code writing with:	1
i. Small but tricky codes	1
ii. Proper parameter passing	1
iii. Command line Arguments	1
iv. Variable parameter	2
v. Pointer to functions	2
vi. User defined header	3
vii. Make file utility	3
viii. Multi file program and user defined libraries	4
ix. Interesting substring matching / searching programs	4
x. Parsing related assignments	4

Lab Course Outcome:

CO 1	Read, understand and trace the execution of programs written in C language.	K2
CO 2	Write the C code for a given algorithm.	K2
CO 3	Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor.	K3
CO 4	Write programs that perform operations using derived data types.	K2
CO5	Implement String Handling	K3

B. Tech.-Second Semester **Branch- CS/ CSE/CSE (R)/ IT/CSE(DS)/CSE(IOT)/CSE(AIML)/CSE(AI)/CYS/ ECE/ECE(VLSI)/ ME/M. Tech (Integrated)/ BT**	
Subject Code-BCSE0251Z	**L - T - P** **0 – 0 - 6**
Subject Name- C Programming	**No. of hours-60**

Course Objective-The objective of a C programming course is to provide students with a solid foundation in the C programming language. The course aims to familiarize students with the syntax, concepts, and principles of C programming, as well as develop their ability to write efficient and effective C code. They will be able to develop complex real-world applications.

Course outcomes: **CO 1:** Implement and trace the execution of conditional and iteration programs. **CO 2:** Implement and trace the execution of conditional and iteration programs.da **CO 3:** Acquire the knowledge of memory allocation and binding, array, structure t o solve complex problems **CO 4:** Compare and contrast between Structure and union along with their applications **CO5:** Develop Complex real-world applications	**K1** **K3** **K3** **K3** **K4**

Course Content

Unit	**Module**	**Topics Covered**	**Pedagogy**	**Lecture Required (T=L+P)**	**Aligned Practical/Asstaignment/Lab**	**CO Mapping**
I	Introduction to Algorithm and C Program	Programming using C: Concepts of Algorithm and Flowchart, Translator and its types, Applications of C programming, Structure of C program, Overview of compilation and execution process in an IDE, transition from algorithm to program, Syntax, logical errors and Run time errors, object and executable code,	T3, R1, Chalk & Duster/PPT/Online Programs	2+2	Basic Program in C	CO1
	Tokens & Operators	Keywords, identifiers,	T3, R1, Chalk & Duster/PPT/Onli	3+3	Basic Program in C	CO1

		constant, data types. Operators and their types, Arithmetic expressions and precedence: Operators, operator precedence and associativity, type conversion, mixed operands	ne Programs			
	Conditional Branching	if, else-if, nested if - else, switch statements, use of break, and default with switch	T3, R1, Chalk & Duster/PPT/Online Programs	1+2	Programs using Conditional Statement	CO1
	Iteration and loops:	Concept of loops, for, while and do-while, multiple loop variables, use of break and continue statements, nested loop.	T3, R1, Chalk & Duster/PPT/Online Programs	1+2	Programs using Looping Statement	CO1
II	Functions:	Concept of Sub-programming, function, types of functions, passing parameters to functions: call by value Definition,	T3, R1, Chalk & Duster/PPT/Online Programs	3+3	Function Programs	CO2
	Recursion	Definition, Types of recursive functions,	T3, R1, Chalk & Duster/PPT/Online Programs	1+2	Recursion Programs	CO2

		Tower of Hanoi problem,				
	Storage:	scope of variable, local and global variables, Nesting of Scope, Storage classes: Auto, Register, Static and Extern	T3, R1, Chalk & Duster/PPT/Online Programs	1+1	Programs showing use of Storage	CO2
	Pointers:	defining and declaring pointer, pointer arithmetic and scaling, Pointer Aliasing. call by reference	R1, R3, R4 Chalk & Duster/PPT/ Labs	2+2	Programs illustrating use of Pointers Arithmetic/Addressing/ Call by Reference	CO2
III	Arraya:	Array notation and representation (one and two dimensional) , array using pointers, manipulating array elements,2-D array s used in matrix computation.	R1, R3, R4 Chalk & Duster/PPT/ Labs	2+2	Programs illustrating use of Pointers Arithmetic/Addressing/ Call by Reference	CO3
	Strings:	Introduction, initializing strings, accessing string elements, Array of strings, Passing	R1, R3, R4 Chalk & Duster/PPT/ Labs	2+3	Use of Arrays both Single and Multi-Dimensional.	CO3

		strings to functions, String functions like Strcat, strcmp, strcpy and any other functions				
IV	Structure:	Introduction, Initializing, defining and declaring structure, accessing members, Operations on individual members, Operations on structures, Structure within structure, Array of structure	T1, T2, R1, R2 Chalk & Duster/PPT/ Labs	2+2	Program Based on structure implementation	CO4
	Union:	Introduction, Initializing, defining and declaring structure, Accessing members, Operations on individual members, Operations on Union, Difference between Structure and Union	T1, T2, R1, R2 Chalk & Duster/PPT/ Labs	1+1		CO4
	Dynamic Memory Allocation	Introduction, Library functions– malloc, calloc, realloc and	T1, T2, R1, R2 Chalk & Duster/PPT/ Labs	1+1	Programs allocating memory during run time and manipulations	CO4

		free.				
V	File Handling	Basics, File Types, File operations, File pointer, File opening modes, File handling functions, Command Line Arguments, File handling through command line argument, Record I/O in files	T1, T2, R1, R2 Chalk & Duster/PPT/ Labs	2+4	Implementation of Data Files and Command Line Arguments	CO5
	Introduction to Embedded Programming	Introduction to Embedded System, Factors for Selecting the Embedded Programming Language, Difference Between C and Embedded C, Keyword, Datatypes, Components of Embedded Program, Program Structure, Basic concepts of Embedded Programming, Defining Macros, Types & File Inclusion, Pre-processor directives implementation	T1, T2, R1, R2 Chalk & Duster/PPT/ Labs	2+4	Example on Embedded Programs	CO5

References-
Textbooks: (T1) Herbert Schildt,"C: The Complete Reference", Osbourne McGrawHill,4thEdition,2002. (T2) Computer Concepts and Programming in C, E Balaguruswami, McGrawHill (T3) Let Us C by Yashwant P.Kanetkar. BPB publication (T4) K.R Venugopal, "Mastering C", TMH (T5) Yashwant P. Kanetkar, "Working with C", BPB publication
Reference Books: (R1) The C programming by Kernighan Brain W.and Ritchie Dennis M.,Pearson Education. (R2) Computer Science-A Structured Programming Approach Using C, by Behrouz A. Forouzan, Richard F. Gilberg, Thomson, Third Edition, Cengage Learning-2007. (R3) Computer Basics and C Programming by V.Rajaraman, PHI Learning pvt. Limited,2015. (R4) Schrum's Outline of Programming with C by Byron Gottfried, McGraw-Hill (R5) Computer Fundamentals and Programming in C.Reema Thareja, Oxford Publication
Links: **E-Book Links:** (E1)https://en.wikibooks.org/wiki/C_Programming (E2)https://en.wikibooks.org/wiki/A_Little_C_Primer (E3) https://www.goodreads.com/book/show/6968572-ansi-c-programming

SYLLABUS

B. TECH.-First Semester **BRANCH- Bio Technology**	
SUBJECT CODE-BBT0101Z	**L - T - P** **3 – 1 - 0**
SUBJECT NAME- ELEMENTRY MATHEMATICS	**No. of hours-42**
Course Objective- The objective of this course is to familiarize the graduate engineers of Biotechnology with techniques in basic algebra, differential calculus, integration and solving Ordinary Differential Equations of first order. It aims to equip the students with standard concepts and tools from basic to intermediate level that will enable them to tackle more advanced level of mathematics and applications that they would find useful in their disciplines.	
Course Outcome – **CO1 -** Apply concept of equation to solve quadratic equations and system of linear inequality in two variables. **CO2-** Apply the concept of differentiation to find the derivative of different type functions, rate of change and maxima and minima. **CO3-** Apply concept of integration to evaluate integrals and definite integrals. **CO4-** Apply the concept of differentiation and integration to find the solution of differential equations. **CO5-** Solve the problems of Profit, Loss, Number & Series, Coding & decoding and Algebra.	
Course Content	

Unit	Module	Topics Covered	Pedagogy	Lecture Required (T=L+P)	Aligned Practical/Assignment/Lab	CO Mapping
Unit 1	Algebra	Statement of Fundamental Theorem of Algebra, solution of quadratic equations in the complex number system. Linear inequalities. Algebraic solutions of linear inequalities in one variable and their representation on the number line. Graphical solution of linear inequalities in two variables. Solution of system of linear inequalities in two variables graphically.	Class room Teaching, Smart Board, PPT, M- tutor.	8	**Assignment 1.1**	CO1
Unit 2	Differential Calculus	Functions, Limit, Continuity and Differentiability. Definition of derivative, physical and geometrical significance of derivative, derivative by first principal. Derivative of sum, difference, product and quotient of functions. Derivatives of polynomial and trigonometric functions, derivative of composite functions, chain rule, derivatives of inverse trigonometric functions, derivative of implicit function. Concept of exponential and logarithmic functions and their derivative. Logarithmic differentiation. Derivative of functions expressed in parametric forms. Second order derivatives. Applications of Derivatives: rate of change, maxima and minima (second derivative test only). Simple problems (that illustrate basic principles	Class room Teaching, Smart Board, PPT, M- tutor.	8	**Assignment-2.1** **Assignment-2.2**	CO2

		and understanding of the subject as well as real life situations).				
Unit 3	Integral Calculus	Integration as inverse process of differentiation. Integration of a variety of functions by substitution, by partial fractions and by parts, only simple integrals of the type to be evaluated. Basic properties of definite integrals and evaluation of definite integrals. Applications of the Integrals: Applications in finding the area under simple curves, especially lines, areas of circles/parabolas/ellipses (in standard form only).	Class room Teaching, Smart Board, PPT, M- tutor.	10	**Assignment-3.1**	CO3
Unit 4	Differential Equations	Definition, order and degree, general and particular solutions of a differential equation. Formation of differential equation whose general solution is given. Solution of differential equations of first order and first degree by method of separation of variables, homogeneous differential equations . Solutions of linear differential equation of the type: $\frac{dy}{dx}+py=q$, where p and q are functions of x.	Class room Teaching, Smart Board, PPT, M- tutor.	8	**Assignment-4.1**	CO4
Unit 5	Aptitude-I	Simplification, Percentage, Profit, loss & discount, Average, Number & Series, Coding & decoding, Algebra.	Class room Teaching, Smart Board, PPT, M- tutor.	8	**Assignment-5.1**	CO5

References-

Text Books:

1. Mathematics - Textbook for Class XI, NCERT Publication

2. Mathematics Part I - Textbook for Class XII, NCERT Publication

3. Mathematics Part II - Textbook for Class XII, NCERT Publication

4. Quantitative Aptitude by R.S. Aggrawal

Reference Books:

1) Higher engineering mathematics by B.V.Ramana (Tata Macgraw Hill)
2) Advanced modern engineering mathemtics by Glyn james (pearson education)

Links:

Unit-1

https://www.youtube.com/watch?v=Ujs30gztM5E

- https://www.youtube.com/watch?v=9MFjoGm06dg
- https://www.youtube.com/watch?v=l1Xri-tvd6g
- https://www.youtube.com/watch?v=NHx_-vE-zQo
- https://www.youtube.com/watch?v=gI3y4OWlLO4

Unit-2

- https://youtu.be/hswdwcNhQ0g
- https://youtu.be/EkkATH3W1Mo
- https://youtu.be/r031pzhBP5c
- https://www.youtube.com/watch?v=lTtsFrkBsOI
- https://www.youtube.com/watch?v=_9MVn-Jw2G4
- https://www.youtube.com/watch?v=HrymMfWU_x0
- https://www.youtube.com/watch?v=dEPr5D6CqQQ
- https://www.youtube.com/watch?v=XzIypjhzj8c
- https://www.youtube.com/watch?v=ZqHPcKq6VNI

Unit-3

- https://www.youtube.com/playlist?list=PLbu_fGT0MPstBzAW5gGWLltksM_yAs3si
- https://youtu.be/z0ajJjA3_Ns

Unit-4

- https://youtu.be/f-4tMNFUqyU
- https://youtu.be/AX_0jNDIi9I
- https://youtu.be/BHdXOPD4cvo
- https://youtu.be/OET0qwat15o

Unit-5

https://www.GovernmentAdda.com

B. Tech.-First Semester **Branch- Bio Technology**	
Subject Code-BBT0102Z	**L - T - P** **3 – 1 - 0**
Subject Name-Remedial Biology	**No. of hours-40**
Course Objective- To introduce students' basic knowledge about structure and function of biomolecules. develop understanding about cell biology, and nucleic acids and understand the morphology and anatomy of plants	
Course Outcome – **CO1 -To understand the basics of living systems K1, K2** **CO 2 To understand key common features of living organisms & its classification K1, K2** **CO 3 To know the anatomy and functions of plants K1, K2** **CO 4 To know the concepts of alleles and genes K1, K2** **CO 5 To understand the plant physiology**	
Course Content	

Unit	Module	Topics Covered	Pedagogy	Lecture Required (T=L+P)	Aligned Practical/Assignment/Lab	CO Mapping
Unit 1	**Cell Biology**	The cell concept, structure of prokaryotic, eukaryotic cells, plant cells and animal cells, Structure and function of cell membrane, cell organelles and their function. Structure and use of compound microscope, Macro and micro molecules, Basic chemical constituents of living body. **Carbohydrates-Classification of carbohydrates, functions of carbohydrates, Lipids-Classification and functions of lipids, Proteins-Structure and functions of proteins, Enzymes- Chemical nature, classification and properties of enzymes, mechanism of enzyme activity**	**Smart board, ppts**	**8**	**NA**	**CO1**
Unit 2	**Classification of living organisms,**	**Classification of living organisms (Five kingdom classification, major groups and principles of classification in each kingdom), Salient features of kingdom Monera, Protista and Fungi. Importance of microbiology, Importance of microorganisms in various fields, Classification and features of microorganisms. - Systematic and binomial system of nomenclature, Concept of animal and plant classification.**	**Smart board, ppts**	**8**	**NA**	**CO 2**

Unit 3	Morphology and anatomy of plants	Tissues in animal and plants, Morphology, anatomy and functions of different parts of plants: Root, stem, leaf, inflorescence, flower, fruit and seed, Concepts of botanical garden, herbaria, zoological park and museums.	Smart board, ppts	8	NA	CO3
Unit 4	. Cell division and Genetics	Concepts of alleles and genes, Mendelian Experiments, Cell cycle (Elementary Idea), mitosis and meiosis, techniques to study mitosis and meiosis	Smart board, ppts	8	NA	CO14
Unit 5	Plant Physiology	Plant Physiology: Concepts of diffusion, osmosis, imbibitions, Movement of water, food, nutrients and gases, Photosynthesis, plant growth and development	Smart board, ppts	8	NA	CO5

References-

Text Books:

1. Biology-Textbook of Class XI, NCERT Publication
2. Biology-Textbook of Class XII, NCERT Publication
3.Together With Biology Study Material for Class 12

Reference Books: Biology 12th Edition by Raven and George Johnson and Kenneth Mason and Jonathan Losos and Tod Duncan. McGrawHill Publications 2. TEXTBOOK OF BIOTECHNOLOGY by PATNAIK, McGraw Hill 3. Basic Biotechnology 3rd Edition by Colin Ratledge& Bjorn Kristiansen, Cambridge University Press

Links: https://www.youtube.com/watch?v=_WM2hJmjctI
https://www.youtube.com/watch?v=ZyWYID2cTK0&t=2s
https://www.youtube.com/watch?v=URUJD5NEXC8&t=28s
Unit 2 https://www.youtube.com/watch?v=qlOOGk7ryxc
https://www.youtube.com/watch?v=b8MfRHfV4Q4
Unit 3 https://www.youtube.com/watch?v=w6yyWyzwqhg
https://www.youtube.com/watch?v=Tl4bQEWN7cQ
https://www.youtube.com/watch?v=cBIGu60gJN0&list=PLKlDmFiIyAnem1SOTmMKXyUy5TDoTVor
Unit 4 https://www.youtube.com/watch?v=EJEd3WhE5-I&t=62s
https://www.youtube.com/watch?v=HyJ86mS2Naohttps://www.youtube.com/watch?v=UD0n3g
fZ0yg

Unit 5 https://www.youtube.com/watch?v=WVaRdAGV11M&t=1111s https://www.youtube.com/watch?v=9tf42ruBr4g&t=969shttps://www.youtube.com/watch?v=9tf42ruBr4g&list=RDCMUCtKAQhsa1D_zKbc3yZmwARQ&start_radio=1&t=1012

B.TECH FIRST YEAR

		L T P	Credits
Course Code	**BCSBS0203Z**	**L T P**	**Credits**
Course Title	**Data Structures and Algorithms**	**3 1 0**	**0**

Course Objectives:

The course covers the basic data structures, algorithm, and efficiency of algorithm, introduction to array, stack, Queue, link list and their implementation. The course aims to give understanding of various searching and sorting algorithms and implementation of tree data structure.

Pre-requisites: Basics of C programming &algorithm

Course Contents / Syllabus

UNIT-I	**Basic Terminologies and Introduction to Algorithm & Data Organization**	**8 hours**

Algorithm specification, Recursion, Performance analysis, Asymptotic Notation - The Big-O, Omega and Theta notation, Programming Style, Refinement of Coding - Time-Space Trade Off, Testing, Data Abstraction

UNIT-II	**Linear Data Structure**	**8 hours**

Array, Stack, Queue, Linked-list and its types, Various Representations, Operations & Applications of Linear Data Structures

UNIT-III	**Non-linear Data Structure**	**8 hours**

Trees (Binary Tree, Threaded Binary Tree, Binary Search Tree, B & B+ Tree, AVL Tree, Splay Tree) and Introduction of Graphs (Directed, Undirected), Various Representations, Operations & Applications of Trees

UNIT-IV	**Searching and Sorting on Various Data Structures**	**8 hours**

Sequential Search, Binary Search, Comparison Trees, Breadth First Search, Depth First Search Insertion Sort, Selection Sort, Shell Sort, Divide and Conquer Sort, Merge Sort, Quick Sort, Heapsort, Introduction to Hashing

UNIT-V	**File & Graph**	**8 hours**

File: Organization (Sequential, Direct, Indexed Sequential, Hashed) and various types of accessing schemes.
Graph: Basic Terminologies, Representations, Operations and Applications of Graphs, Graph search and traversal algorithms and complexity analysis.

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

CO1	Analyzeand implement arrays, linked lists, stacks, queues to solve complex problems.	K3, K4
CO2	Compare the computational efficiency of the sorting and searching algorithms.	K4
CO3	Assessthe memory representation oftree and perform various operations on these data structure.	K3
CO4	Apply the concept of recursion to solve the real-world problems.	K3
CO5	Develop the algorithms using graph data structures.	K6

Text Books

1. E. Horowitz, S. Sahni, S. A-Freed, Fundamentals of Data Structures, 2008, Universities Press.
2. A. V. Aho, J. E. Hopperoft, J. D. UIlman,Data Structures and Algorithms, 1983, Pearson.

Reference Books

1. Donald E. Knuth, The Art of Computer Programming: Volume 1: Fundamental Algorithms, 1968, Addison-Wesley.
2. Thomas, H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 2009, 3rd Edition, The MIT Press.
3. Pat Morin,Open Data Structures: An Introduction (Open Paths to Enriched Learning), 2013, 31st Edition, UBC Press.

B.TECH FIRST YEAR

Course Code	BCSBS0253Z	L T P	Credit
Course Title	Data Structures and Algorithms Lab	0 0 4	0

Suggested List of Experiments	CO
1. Program to create and display linear array	CO1
2. Program to insert a data item at any location in a linear array	O1
3. Program to delete a data item from a linear array	CO1
4. Program to implement linear search in an Array	CO1
5. Program to implement binary search in the sortedarray without recursion	CO1, CO4
6. Program to implement binary search in the sortedarray with recursion	CO1, CO4
7. Program to implement bubble sort in a non-recursive way	CO1, CO4
8. Program to implement selection sort in a non-recursive way	CO1, CO4
9. Program to implement insertion sort in a non-recursive way	CO1, CO4
10. Program to implement merge sort in a non-recursive way	CO1, CO4
11. Program to implement merge sort in a recursive way	CO1, CO4
12. Program to implement Queue Using array	CO1, CO3
13. Program to implement Circular Queue Using array	CO1, CO3
14. Program to implement Stack Operation using array	CO1, CO3
15. Program to implement the Single Linked List a. Insertion b. Deletion c. Traversal d. Reversal e. Searching f. Updation g. Sorting h. Merging	CO1
16. Program to implement the doubly Linked List a. Insertion b. Deletion c. Traversal d. Reversal e. Searching f. Updation g. Merging	CO1
17. Program to implement the circularly Single Linked List a. Insertion b. Deletion c. Traversal d. Reversal e. Searching f. Updation	CO1
18. Program to implement Queue Using linked list	CO1, CO3
19. Program to implement Circular Queue Using linked list	CO1, CO3
20. Program to implement Priority Queue Using linked list	CO1, CO3
21. Program to implement Stack Operation using Linked list	CO1, CO3
22. Program to implement Tower of Hanoi	CO2
23. Program implementing Addition of two polynomials via Linked Lists	CO1
24. Program to implement binary tree using linked list a. Insertion b. Deletion c. Traversal d. Searching	CO1, CO5
25. Program to implement binary search tree using linked list	CO1,

a. Insertion b. Deletion c. Traversal d. Searching		CO5
26. Program to implement heap sort in a non-recursive way		CO1, CO4
27. Program to implement BFS algorithm		CO5
28. Program to implement DFS algorithm		CO5
29. Program to implement the minimum cost spanning tree		CO5
30. Program to implement the shortest path algorithm		CO5
Lab Course Outcome: At the end of course, the student will be able to		
CO1	Write programs for solving mathematical problems using array and linked list.	K3
CO2	Implement concept of recursion to solve complex problem.	K3
CO3	Implement various operations of stack and queue data structure.	K3
CO4	Write efficient sorting, searching programs.	K3
CO5	Implement program to solve real world problem using tree and graph data structure.	K3