

## COURSE OUTLINE FOR CERTIFICATE COURSE “JUNIOR SOFTWARE DEVELOPER”

<b><i>Name of Certificate Course</i></b>	<i>Junior Software Developer</i>
<b><i>NSQF Level</i></b>	4
<b><i>Eligibility</i></b>	Students pursuing undergraduate programme in Science Stream
<b><i>Batch Size</i></b>	30

**TABLE-I (12 CREDITS TO BE OFFERED IN EMBEDDED MODE IN 3<sup>RD</sup>/4<sup>TH</sup>, 5<sup>TH</sup> & 6<sup>TH</sup> SEMESTER)**

<b><i>Course Code/Course Title</i></b>	<b><i>Credit Weightage</i></b>			<b><i>Duration in Number of Hours</i></b>
	<b><i>Theory</i></b>	<b><i>Practical</i></b>	<b><i>Total</i></b>	
<b><i>JSD-1</i></b> Programming with C (Basic)	2	2	4	90
<b><i>JSD-2</i></b> Programming with Python	2	2	4	90
<b><i>JSD-3</i></b> Advanced Python Programming	2	2	4	90

To facilitate forward linkage of skill courses of 12 credits indicated in Table-I with the earning of UGC/NSQF skill certificate, add-on courses of 18 credits as indicated in Table-II are available on optional basis.

**TABLE-II (18 CREDITS ADD-ON COURSES FOR CERTIFICATE COURSE “JUNIOR SOFTWARE DEVELOPER”)**

<b><i>Course Code/Course Title</i></b>	<b><i>Mode of training</i></b>	<b><i>Venue of training</i></b>	<b><i>Credit Weightage</i></b>	<b><i>Duration in Number of Hours</i></b>	<b><i>Semester in which to be offered</i></b>
<b><i>JSD-4</i></b>	Three Weeks Industrial Workshop-I	To be decided in consultation with Mentor Institution	6	90	These credits can be covered during winter vacations/ or after semester-end exams depending upon the availability of time
<b><i>JSD-5</i></b>	Three Weeks Industrial Workshop-II	To be decided in consultation with Mentor Institution	6	90	
<b><i>JSD-6</i></b>	Three Weeks Industrial Internship	To be decided in consultation with Mentor Institution	6	90	

**1<sup>st</sup> SEMESTER**  
**COMPUTER APPLICATIONS**  
**(JUNIOR SOFTWARE DEVELOPER)**  
**SKILL ENHANCEMENT COURSE (SEC)**

**JSD122S: PROGRAMMING WITH C (BASIC)**

**CREDITS: THEORY: 2, PRACTICAL: 2**

**THEORY (2 CREDITS)**

**UNIT 1 – C LANGUAGE FUNDAMENTALS (15 LECTURES)**

Introduction to Programming, Compilers, Interpreters and Assembler. Algorithm and Flowchart, Introduction of C Language. Reserved Words, Built-In Data Types, Variables, Operators and Expressions, Statements, Compound Statements. Using Standard Inputs and Output Functions (printf, scanf). Editing Compiling & Linking a Program. The C-preprocessor and its use in Macro Definition, Operators: Assignment, Arithmetic, Relational, Logical, Conditional and Assignment Operator. Increment & Decrement (pre & post) Operators, Bitwise Operators. Control Statements: If, else & Switch-Case.

**UNIT 2 – CONTROL STATEMENTS, LOOPS, ARRAY AND FUNCTIONS (15 LECTURES)**

Loop Statements: for, while, do while (with break & continue) Statements. Nested Loops.

Arrays: Array Initialization, Multi-Dimensional Arrays; Character Arrays & Strings; String Processing Functions.

User-Defined Functions: Prototype & Definition; Parameter Passing. Recursive Functions. Scope & Lifetime of Variables.

**Reference Books:**

1. Balaguruswamy, Programming in ANSI C, Tata McGraw Hill.
2. Torrence W Pratt, Programming Language Design and Implications, PHI.
3. Gottfried Programming with C.
4. Let Us C by Yashwant Kanetkar BPB Publications.

## PRACTICALS (CREDITS: 2)

### LAB SHEET-PROGRAMMING WITH C (BASIC)

1. Write a program in C to read value of variable and display it,
2. Write a program in C to find sum of two numbers entered by the user.
3. Write a program in C to demonstrate the use of arithmetic operators.
4. Write a program in C to demonstrate use of relational operators.
5. Write a program in C to demonstrate the use of increment and decrement (pre and post) operators.
6. Write a program in C to check whether a number entered by user is odd or even.
7. Write a program in C to find largest of three numbers.
8. Write a program in C to display first  $n$  natural numbers.
9. Write a program in C to find sum of first  $n$  natural numbers.
10. Write a program in C to compute factorial of a number.
11. Write a program in C to compute Fibonacci series.
12. Write a menu driven program in C to perform addition, subtraction and multiplication of two numbers. Make use of switch-case statement
13. Write a program in C to demonstrate use of conditional operator.
14. Write a program in C to display individual digits of a number.
15. Write a program in C to compute sum of digits of a number.
16. Write a program in C to reverse a number.
17. WAP to compute the sum of the first  $n$  terms of the following series  
 $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
18. WAP to compute the sum of the first  $n$  terms of the following series  
 $S = 1 - 2 + 3 - 4 + 5 - \dots$
19. Write a program in C to demonstrate argument passing *by value* and *by reference*
20. Write a function in C to compute factorial of a number.
21. Write a recursive function in C to compute factorial of a number.
22. Write a program in C to demonstrate use of basic Math functions inbuilt in C
23. Write a program in C to create an array, read its elements from user, traverse / display the individual elements.
24. Write a program in C to compute sum of elements of an array
25. Write a program in C to check whether an element exists in an array or not.
26. Write a program in C to create a two dimensional array, read its elements, traverse/display the individual elements.
27. Write a program in C to find sum of two matrices
28. Write a program in C to compute multiplication of two matrices.
29. Write a program in C to demonstrate basic String functions.
30. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not