

UNIVERSITY OF DELHI

Multidisciplinary Courses of Study in Computer Science with Two Core Disciplines

(SEMESTER-I)

based on

Undergraduate Curriculum Framework 2022 (UGCF)

(Effective from Academic Year 2022-23)



University of Delhi

DSC1:- Introduction of Programming using C++

Course Title	Nature of the Course	Total Credits	Components			Eligibility Criteria/ Prerequisite	Contents of the course and references may be seen at
			L	T	P		
Introduction of Programming Using C++	DSC-1	4	3	0	1	Class XII Pass	Annexure – I

Discipline-A1- Programming fundamentals using Python

Course Title	Nature of the Course	Total Credits	Components			Eligibility Criteria/ Prerequisite	Contents of the course and references may be seen at
			L	T	P		
Programming Fundamentals using Python	A1	4	3	0	1	Class XII Pass	Annexure – II

Department of Computer Science

University of Delhi

Curriculum

Multidisciplinary Courses of Study with Two Core Disciplines

under UGCF 2022

University of Delhi

Approved in

UG Committee meeting held on May17, 2022

Faculty of Mathematical Sciences meeting held on May 25, 2022

Standing Committee on Academic Matters

1. [Table of DSCs](#)
2. [Table of Discipline Courses](#)

Table of DSCs

Semester	DSC -No.	Title	L	T	P	Total credits	Prerequisite
I	DSC 01	Introduction to Programming using C++	3	0	1	4	Pass in Class XII
II	DSC 02	Data structures	3	0	1	4	DSC 01
III	DSC 03	Computer System Architecture	3	0	1	4	Pass in Class XII
IV	DSC 04	Operating systems	3	0	1	4	DSC 01 , DSC 02
V	DSC 05	Database Management System II	3	0	1	4	Pass in Class XII
VI	DSC 06	Computer Networks	3	0	1	4	DSC 01 /a course in C/C++ at plus 2 level/** DSC 02 DSC 04
VII	DSC 07	Design and Analysis of Algorithms	3	1	0	4	DSC 02

VIII	DSC 08	Information Security	3	0	1	4	DSC 01 DSC 02 DSC 03 DSC 04 DSC 05 DSC 06 DSC 07
------	------------------------	----------------------	---	---	---	---	--

Table of Discipline Courses

Semester	Discipline	Title	L	T	P	Total credits	Pre requisites
I	A1	Programming Fundamentals Using Python	3	0	1	4	Pass in Class XII
II	A2	Data Visualization using Python	2	0	2	4	A1
III	A3	Database Management System I	3	0	1	4	Pass in Class XII
IV	A4	Introduction to Web Programming	3	0	1	4	Pass in Class XII
V	A5	Multimedia and its Applications	2	0	2	4	Pass in Class XII
VI	A6	Introduction to Cyber Security (under preparation)	3	0	1	4	Pass in Class XII

Note: Batch size for Practicals will be (8-10) and Tutorials will be (12-15).

DSC 01: Introduction to Programming Using C++

Course Objective

The course introduces Object Oriented Programming Language C++ with the objective to use object oriented features to develop efficient programs.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Use the built-in data types of C++ programs and write and execute a simple program
2. Apply selection and iteration to solve programming problems.
3. Understand and implement arrays and user defined functions.
4. Comprehend the concepts of Object Oriented programming.

Unit 1 Introduction to C++: Need and characteristics of Object-Oriented Programming, Structure of a C++ Program (main () function, header files, output, input, comments), compile and execute a simple program.

Unit 2 Data types and Expressions: Keywords, built in data types, variables and constants, naming convention, Input-Output statements, operators and their precedence, expressions, typecasting, library functions.

Unit 3 Control Constructs in C++: Decision making using selection constructs, iteration using looping constructs.

Unit 4 Arrays, Pointers and User defined functions: Defining and initializing single and multi-dimensional arrays, user defined functions, passing arguments to functions, returning values from functions, inline functions, default arguments, introduction to pointers.

Unit 5 Classes and Objects: Need and implementation of abstraction creating classes, objects as function arguments, modifiers and access control, constructors and destructors.

References

1. E. Balaguruswamy, *Object Oriented Programming with C++*, 7th edition, McGraw-Hill Education, 2017.
2. Robert Lafore, *Object Oriented Programming in C++*, 4th edition, SAMS Publishing, 2008.

Additional References

- (i) D.S. Malik, *C++ Programming: From Problem Analysis to Program Design*, 6th edition, Cengage Learning, 2013.
- (ii) Herbert Schildt, *C++: The Complete Reference*, 4th Edition, McGraw Hill, 2003.

Suggested Practical List

1. Write a program to find the largest of n natural numbers.
2. Write a program to find whether a given number is prime or not.
3. Write a program that takes a positive integer n and the produce n lines of output as shown:

```
*  
* *  
* * *  
* * * *
```

(for n = 4)

4. Write a menu driven program for following:
 - a. to check whether a given number is odd or even.
 - b. display a fibonacci series
 - c. compute factorial of a number
5. Write a program to accept a number, reverse it and print the sum of its digits.
6. Write a program using functions to print the series and its sum:

$$1 + 1/2! + 1/3! + \dots + 1/n!$$

7. Write a program to perform the following operations on an input string
 - a. Print length of the string
 - b. Find frequency of a character in the string
 - c. Print whether characters are in uppercase or lowercase
 - d. to check whether a given string is palindrome or not.
8. Write a program that will prompt the user for a list of 5 prices. Compute the average of the prices and find out all the prices that are higher than the calculated average.
9. Design a class named Vehicle, having registration number and year as its private members. Define a suitable constructor and a method to print the details of a vehicle. Write a C++ program to test the above class.
10. Inherit a class Car from the Vehicle class defined above. Add model to the Car class. Define a suitable constructor and a method to print the details of a car. Write a C++ program to test inheritance of this class.

A1 : Programming Fundamentals Using Python

Course Objective

This course is designed as the first course that introduces programming concepts using Python to the students. The course focuses on the development of Python programming to solve problems of different domains

Course Learning Outcomes

On successful completion of this course, a student will be able to:

1. Understand the basics of programming language
2. Develop and document modular Python programs.
3. Apply suitable programming constructs and built-in data structures to solve a problem.

Syllabus

Unit 1 Introduction to Python Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Python interpreter/shell, indentation; Executing simple programs in Python.

Unit 2 Creating Python Programs: Identifiers and keywords; literals, numbers, and strings; Operators and expressions; Input and output statements; control structures (conditional statements, loop control statements, break, continue and pass), Errors and exception handling.

Unit 3 User Defined Functions: Defining functions, passing arguments and returning values, default arguments

Unit 4 Built-in data structures: Strings, Lists, Tuples, Sets, Dictionaries; their built-in functions, operators and operations.

References

1. Kamthane, A. N., & Kamthane, A.A. *Programming and Problem Solving with Python*, McGraw Hill Education, 2017.
2. Balaguruswamy E. “*Introduction to Computing and Problem Solving using Python*”, 2nd edition, McGraw Hill Education, 2018.
3. Taneja, S., Kumar, N. *Python Programming- A modular Approach*, Pearson Education India, 2018.

Additional References

- (i) Guttag, J. V. *Introduction to computation and programming using Python*, MIT Press, 2018.
- (ii) Downey, A. B. *Think Python–How to think like a Computer Scientist* 2nd edition. O’Reilly 2015.

Suggested Practical List

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:
Grade A : if Percentage ≥ 80
Grade B : if Percentage ≥ 60 and Percentage < 80
Grade C : if Percentage ≥ 40 and Percentage < 60
Grade D : if Percentage ≤ 40

2. WAP to print factors of a given number.
3. WAP to add N natural numbers and display their sum.
4. WAP to print the following conversion table (use looping constructs):

Height(in Feet)	Height(in inches)
5.0ft	60 inches
5.1ft	61.2inches

5.8ft	69.6inches
5.9ft	70.8inches
6.0ft	72inches

5. WAP that takes a positive integer n and the produce n lines of output as shown:

*

**

(for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.
7. Write a function that calculates factorial of a number n.
8. WAP to print the series and its sum: (use functions)

$$1/1! + 1/2! + 1/3! + \dots + 1/n!$$

9. WAP to perform the following operations on an input string
 - a. Print length of the string
 - b. Find frequency of a character in the string
 - c. Print whether characters are in uppercase or lowercase
10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple $t1 = (1,2,5,7,2,4)$. The program should perform the following:
 - a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.
 - b. Concatenate tuple $t2 = (10,11)$ with $t1$.