

Roll No.

--	--	--	--	--	--	--	--	--	--	--	--

Total No. of Pages : 02

Total No. of Questions : 07

BCA (Sem.-2)

OBJECT ORIENTED PROGRAMMING USING C++

Subject Code : UGCA-1909

M.Code : 77417

Date of Examination : 14-12-2024

Time : 3 Hrs.

Max. Marks : 60

INSTRUCTIONS TO CANDIDATES :

1. SECTION-A is COMPULSORY consisting of TEN questions carrying TWO marks each.
2. SECTION-B contains SIX questions carrying TEN marks each and students have to attempt any FOUR questions.

SECTION-A

1. Explain the following:

- i) Name any three preprocessor directives in C++.
- ii) List the features of object oriented programming language C++.
- iii) What are the steps involved in execution of a C++ program?
- iv) What is member functions in a class?
- v) How many types of access specifiers are provided in C++?
- vi) What is dynamic initialization of Objects in C++?
- vii) Differentiate between multiple and multilevel inheritance.
- viii) What are rules for overloading operators?
- ix) What is an abstract class?
- x) What are the various file streams available in C++?

SECTION-B

2.
 - i) Write a sample code to show the structure of C++ program code.
 - ii) What are the difference between procedure oriented languages and object oriented programming language?
3. What is class and object? How a class is declared? Explain concept of array of objects by taking a suitable example.
4. Explain the use of constructors and destructors in C++ with the help of an example. Can we use multiple constructors in a class in C++.
5. What is inheritance? Explain different types of inheritance in C++. Write a program to show use of single inheritance.
6.
 - i) What do you mean by operator overloading?
 - ii) Explain the concept of early binding and late binding in detail.
7. What are the various methods of opening files in C++? How to read and write data to files in C++? Explain giving examples.

NOTE : Disclosure of Identity by writing Mobile No. or Marking of passing request on any paper of Answer Sheet will lead to UMC against the Student.