

[This question paper contains 4 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 6348

**K**

Unique Paper Code : 2342571102

Name of the Paper : Programming Using C++

Name of the Course : **B.A. (P) CS M/N.M. & B.Sc. (PS/MS) with CS**

Semester / Type : I / Discipline Specific Course A1 DSC

Duration : 3 Hours

Maximum Marks : 90

**Instructions for Candidates**

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **Section A** is compulsory.
3. Attempt any **four** questions from **Section B**.
4. Parts of a question must be answered together.
5. Use of scientific calculator is allowed.

**Section A: 30 Marks**

1. a) How built-in data types are different from derived data types. Give example. 5
- b) Which of the following are valid identifiers and which of the following 5  
invalid identifiers? Give reasons for each.
  - i) @count
  - ii) \_count
  - iii) 3count
  - iv) #marks
  - v) Marks#
  - vi) english\_marks
  - vii) countofstudents
  - viii) gmg4fuzj
  - ix) wluqfvy\$
  - x) ge7z4z
- c) Answer the following in one word: 5
  - i) An object-oriented programming language which does not allow multiple inheritance.

*P.T.O.*

- ii) The process of making an operator exhibit different behaviour in different instances is known as ?
- iii) Header file that contains macros and helps in debugging.
- iv) Header file that contains classes for identifying data types during execution.
- v) Header file that contains classes for stream manipulation and formatting of data streams.
- d) Explain under what circumstances the following statements are used (any two) 2.5+2.5
  - i) throw;
  - ii) void fun1(float x) throw()
  - iii) catch (...)
- e)
  - i) Give four instances where inline function may not work. 2
  - ii) Explain with example if a compiler ignore the inline function definition ? 1
  - iii) Explain the tradeoff associated with the inline function ? 2
- f) How multilevel inheritance and multiple inheritance are different? Explain with example. 5

### Section B: 60 Marks

2. a) Explain with example how pointer to functions is different from pointer to objects? 5
- b) Write a C++ program that uses a for loop to compute the sum of the first n natural numbers. The program asks the user to input a value for n and print out the sum. 6
- c) Explain the following statements with the help of an example: 2+2
  - i) break
  - ii) continue
3. a) Find the error and correct the code 2\*5
  - i) `int m==40;`
  - ii) `if(x = 10) cout << "Yes";`
  - iii) `float arr[5] = {1, 2, 3, 4, 5, 6};`
  - iv) `int *p = &x; delete p;`
  - v) `for(int i=0; i<10; i--);`
- b) Why are access specifiers (private, public, protected) considered a tool of encapsulation? 5
4. a) Write a menu driven C++ program simulating a simple Shape hierarchy. Declare a base class called Shape with a virtual function area(). Derive two classes, Circle and Rectangle, each overriding the area() function to calculate the area of a circle and a rectangle respectively. The program will take all inputs necessary-for example, the radius for the circle and the length and width for the rectangle. The program should calculate the area and print the results. 8

- b) Explain call-by value and call-by reference with the help of example. Write a program to swap two numbers using these two concepts. 4+3
5. a) Explain how polymorphism can be achieved during run time using virtual functions. 5
- b) Write a program using c++ for the following numeric pyramid pattern: 5

```

      1
    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

```

- c) Given an array of  $n$  integers, write a user interactive program in C++ using recursive function to print all possible subsets of the array. The program should take array size and array values from the user. 5
6. a) Differentiate between object-based programming language and object-oriented language. 6
- b) Identify the error and rectify. Explain the reason also. 3\*3

**i) Code 1**

```

#include <iostream>
using namespace std;
int main() {
    int *ptr;
    *ptr = 10;
    cout << "Value: " << *ptr << endl;
    delete ptr;
    return 0;
}

```

**ii) Code 2**

```

#include <iostream>
using namespace std;
class Test {
    int x;
public:
    Test(int val) { x = val; cout << "Constructor
called\n"; }
    ~Test() { cout << "Destructor called for "
<< x << endl; }
};

```

```
int main() {  
    Test t1(5);  
    {  
        Test t2(10);  
    }  
    t2.~Test(); // Explicit destructor call  
    return 0;  
}
```

**iii) Code 3**

```
#include <iostream>  
using namespace std;  
class Account {  
private:  
    double balance;  
public:  
    void setBalance(double b) { balance = b; }  
};  
int main() {  
    Account obj;  
    obj.balance = 1000.0;  
    cout << "Balance set successfully!" << endl;  
    return 0; }
```

7. Differentiate between the following terms (any five):

3\*5

- i) Objects and classes
- ii) Inheritance and polymorphism
- iii) Early binding and Late Binding
- iv) Data Abstraction and Data Encapsulation
- v) Operator overloading and function overloading
- vi) ifstream and ofstream