COMP – 3-2 (RC)

S.E. (Comp.) (Semester – III) (RC) Examination, Nov./Dec. 2012
BASICS OF C++

Duration : 3 Hours | Max. Marks : 100

Instructions : *1) Attempt **any five** questions by selecting at least **one** from **each** Module.*
*2) Assume **suitable** data **if** necessary.*

MODULE – I

1. a) Explain with examples the conversion from basic type to class type and class type to basic type. **5**

   b) What are the benefits of object oriented programming ? **2**

   c) Explain the need of new and delete operators in C++. **5**

   d) Enlist the salient differences between C and C++. **8**

2. a) Explain the importance of data encapsulation in C++. **4**

   b) Consider the following two statements : **8**

   ```
   int x = 15, y = 12, z = 6, m ;
   m = (x + y)*z;
   ```

   Using precedence of operators explain the execution of the expression.

   c) Write code to demonstrate the difference between the following : **8**

   i) Break and continue statement

   ii) Gets and puts

MODULE – II

3. a) Write a function to add two numbers using pointers. **8**

   b) Write code to input an array and compute the transpose. **8**

   c) What is the difference between a C++ pointer and a reference ? **4**

P.T.O.

4. a) Write a program to search an Array using Binary Search. 8

   b) Give an example to explain what is a recursive function ? What are the advantages and disadvantages of recursive functions ? 6

   c) What is the difference between local, static and global variables ? Give an example. 6

MODULE – III

5. a) Write a C++ program to read and process students data using an array of structures. Each student data should include student name, roll number, date of birth and marks scored. Read the data for *n* students and print the name and roll number of the topper. 8

   b) What do you mean by function overloading ? Explain with an example. 4

   c) Explain the use of constructors and destructors in C++ with a suitable example. 6

   d) What is the output of the following prcgram ? Justify your answer. 2

```
#include <iostream.h>
struct outer
{
int a ;
int b ;
struct inner
      {
int c ;
int d;
      } ;
};
int main ()
{
cout << " Size of outer is" << size of (outer) << endl;
return 0 ;
}
```

6. a) Explain enumerated data type in C++ with an example. 4

   b) A programmer is writing a function that requires four input values, but most of the time when the function is called, three of the inputs are the same (only changed occasionally). Explain how the programmer should design this function so that it can be called without always passing the three nearly constant values. 3

c) What are inline functions ? Write an inline function for finding the perimeter of a rectangle. 5

d) Write a C++ class to implement a vector of integers. Include member functions for creating the vector and displaying the vector. Overload the [ ] operator for modifying the value of an element in the vector and the * operator for multiplying the vector by a scalar value. 8

MODULE – IV

7. a) State and explain the various class relationships giving suitable examples. 6

b) What is multiple inheritance ? Explain the syntax of multiple inheritance in C++. 4

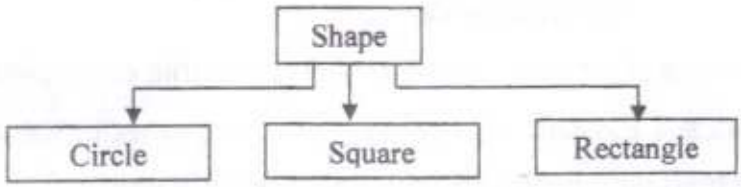
c) What is the output of the following program ? Explain your answer. 4

```
#include <iostream.h>
class Base
{
int x;
public :
Base () { x = 0; cout << "Default base \n" ;}
Base (int y) {x = y; cout <<"Overloaded base\n";}
Void Show () { cout << "Base shows " << x <<endl;}
} ;
class Derived : public Base
{
int y ;
public :
Derived () {y = 0 ; cout << "Default derived \n"; }
Derived (int x) {y = x; cout << "Overloaded derived \n" ;}
void Show () { cout << "Derived shows" << y <<endl;}
} ;
int main ()
{
  Derived obj (4) ;
 Base *obj_ptr = & obj ;
      obj_ptr->Show();
return 0;
}
```

d) Explain the exception handling model in C++. 6

8. a) Explain the concept of polymorphism in object-oriented programming. How does C++ support polymorphism ? Explain. 6

b) Write C++ classes to implement the following inheritance. Provide functionality to read the dimensions and calculate the area for each shape.



8

c) Explain the dynamic memory allocation features in C++. Write the C++ code to allocate and free memory for an array of integer pointers. 6