



COMP 3-2 (RC)

S.E. (Comp.) (Sem. III) (RC) Examination, May 2010

BASICS OF C++

Duration : 3 Hours

Total Marks : 100

Instructions : 1) Answer **any five** questions by selecting at least **one** question from **each** Module.

2) Write the code using C++ language.

3) Make **appropriate** assumptions wherever necessary.

MODULE - I

1. a) List the differences between structured design and object oriented design. 4
- b) Write a C++ program to print multiplication table upto the table of 5 using do while loop. 6
- c) What are variables? List C++ rules for naming variables. 5
- d) Give output of the following code: 2

```
int a = 0;  
do  
{  
    cout << " / n Blue ";  
    ++a;  
} while (a < 4);
```
- e) Explain the meaning of object oriented design. 3
2. a) Briefly explain Johnstons Rules for programmers. 6
- b) Give output of the following code. 3

```
main ()  
{  
    int i = 12, j = 5;  
    float f;  
    f = i/j;  
    cout << " i/j = " << f << endl;  
    f = (float) i/j;  
    cout << " (float) i/j = " << f << endl;  
    f = i / (float) j;  
    cout << " i / (float) j = " << f << endl;  
}
```

P.T.O.

- c) Explain the various control structures and loops in C++ use flowcharts and examples. 6
- d) What is an operator precedence ? List all the operators used in C++ according to their precedence and mention their associativity. 5

MODULE – II

3. a) What are functions ? Write an function for finding minimum of two numbers. 6
- b) Explain the use of "size of" operator in C++ with an example. 5
- c) Explain how a two dimensional array is initialized with a integer value 5 ? Use a diagram and a code to show the initialization. 4
- d) What are default arguments ? How are actual arguments different from formal arguments? 3
- e) Give the advantages of using pointer in C++ program. 2
4. a) Differentiate between call by value and call by reference. 5
- b) Write a C++ program that reads a integer value from one file, another integer value from second file. Adds the two values and stores the result in third file. 6
- c) Explain the three basic statements required for every functions in C++ using an example. 4
- d) Explain how a one dimensional array can be passed to a function ? 5

MODULE – III

5. a) Is overloading pre increment ++ and post increment ++ same. Justify your answer. 4
- b) What are the advantages of using new operator as compared to malloc () ? 3



- c) Write a class to represent a vector. Include constructors and member functions to perform following functions : 8
- i) To create vector
 - ii) Modify value of given vector
 - iii) Multiply by scalar value
 - iv) Add two vectors.
- d) Which are different access specifier in C++ ? Give their importance and example for each. 5
6. a) Write a program to overload +, <=, >=, == operator for string data type. 8
- b) Write program for processing students details using nested structure. 8
- c) What are default arguments ? Give two examples. 4

MODULE - IV

7. a) What is pure virtual function ? How it is defined? 4
- b) What is virtual base class ? Explain. 4
- c) Write a program to take input two numbers and throw divide by zero exception otherwise compute value. 6
- d) Write program to show use of pointer to 3 dimensional array. 6
8. a) What are advantages of inheritance ? Explain different types of inheritance. 7
- b) Consider example of book shop which sells books and video tapes. This two classes are inherited from the base class called media. In base class define member functions which are virtual. Write a program which models the class hierarchy for book shop and processes objects of these classes using pointers to the base class. 8



c) What is output of following code

5

```
#include <iostream.h>
```

```
class base { public :
```

```
    virtual void funct 1 ()
```

```
    { cout << endl << "Base function called"; }
```

```
};
```

```
class derived 1 : public base
```

```
{ public : void funct 1 ()
```

```
    { cout << endl << "Derived function 1 called"; }
```

```
    virtual void funct 2 ()
```

```
    { cout << endl << "Derived class function 2 called"; }
```

```
};
```

```
class derived 2 : public derived 1
```

```
{ public : void funct 1 ()
```

```
    { cout << endl << "derived 2 function"; }
```

```
    void funct 2 ()
```

```
    { cout << endl << "derived 2 function 2 called"; }
```

```
};
```

```
int main ()
```

```
{ base * ptr ; derived 1 * ptr 2 ; base b ;
```

```
    derived 2 d ; ptr = &b ; ptr 2 = &d ;
```

```
    ptr → fun 1 () ; ptr 2 → funct 1 () ;
```

```
    (( derived * ) ptr) → funct 1 () ;
```

```
    return 0 ;
```

```
}
```