



DOON UNIVERSITY, DEHRADUN
School of Technology

Integrated MCA, Second Semester, Mid Semester Examination, 2016
STM-513: Object Oriented Programming (C++)

Time Allowed: 2 Hours

Maximum Marks: 30

SECTION A (Short Answer Type Questions)

(Total Marks: 06)

1. State True/False. (2)

- a) A friend function can access the protected members of the class.
- b) Every function can be inlined.
- c) Constructors can't be overloaded.
- d) A static data member exists before any object of its class is created.

2. What output does the following program produce? Give the order of various function calls. (2)

```

class test {
    int id;
public:
    test(int x) { id = x; cout << "\nConstructing " << id; }
    ~test() { cout << "\nDestructing " << id; }
    void setId(int x) { id = x; }
    int getId() { return id; }
};

void fun(test o) {
    o.setId(2);
    cout << endl << o.getId;
}

int main() {
    test ob(1);
    fun(ob);
    cout << endl << ob.getId;
    return 0;
}

```

3. Choose the correct option(s). (2)

- a) The new operator can be used to allocate memory for
 - A) an array of objects
 - B) a single object
 - C) an array of pointers
 - D) an array of references
- b) Which of the following are inherited by a derived class?
 - A) friends
 - B) constructor and destructor
 - C) private members
 - D) none

- c) Which of the following is used to keep the call by reference value as intact?
- A) enum
 - B) static
 - C) const
 - D) #define
- d) What is the default visibility mode for class members?
- A) private
 - B) protected
 - C) public
 - D) none

SECTION B (Long Answer Type Questions)

(Total Marks: 4 x 3 = 12)

1. a) Define *friend* function. Mention two circumstances, with examples, where friend functions are needed.
b) Explain the concept of *this* pointer.
2. a) Define *inline* function. How do you inline class member functions explicitly?
b) How are static data members of a class defined? Explain the usage of static members using a suitable example.
3. a) Compare function overloading and overriding with examples.
b) What is an *abstract* class? How are they useful?
4. Consider the class *point* with the following two private data members:


```
int x;           // x-coordinate
int y;           // y-coordinate
```

 Define the following overloaded operators:
 - a) prefix decrement operator that would decrement *x* and *y* by 1 each.
 - b) postfix decrement operator that would decrement *x* and *y* by 1 each but return the original value.
 - c) compound assignment operator +=.

SECTION C (Very Long Answer Type Questions)

(Total Marks: 3 x 4 = 12)

1. Comment upon the statement, "Default arguments can be used as a shorthand form of function overloading."
2. Explain any two of the following aspects of C++ language. Use short code fragments to complement your explanation:
 - a) Inheritance and virtual functions
 - b) Overloaded functions and operators
 - c) Exception handling
3. a) State the necessity of defining a copy constructor for a C++ class? Define a copy constructor for a class *String* that uses following two private data members:


```
char *p;        // p points to the first character of the string
int len;        // len holds the length of the string
```

 b) Why might it be useful to define the assignment operator for a C++ class? Define overloaded assignment operator to assign a string/*String* object to a *String* object.