# W3101 Programming Languages – C++ Midterm exam Oct 01, 2008

Name: Student Id:

---

1. Explain the following (OOP) features briefly and show how they are implemented in C++, using ONE example. That is, use a single example (say a base class and a child class) to explain all the features. . . . (4 marks)

   Example:

   (a) Data encapsulation

   (b) Inheritance with public, private and protected members

(c) Polymorphism

(d) Virtual functions and pure virtual functions

(e) Abstract classes

(f) Function overloading

(g) Passing parameters by reference

(h) Constructor and Destructor

2. Answer “True” or “False” to the following questions with explanation. . . . (2 marks)

   (a) C++ programs form a superset of C programs in terms of the syntax, declarations and language specific constructs.

   (b) Complex C++ programs can not logically be implemented in C programming language because OOP specific features and other C++ related features are not supported in C language.

3. Consider the following code segment: . . . (3 marks)

```
class baseClass
{
  public:
    baseClass()
      { cout << ‘‘In base class constructor’’ << endl; }
   ˜baseClass()
```

```
      { cout << ''In base class destructor'' << endl; }
    virtual void f1()
      { cout << ''In base class f1'' << endl; }
    void f2() { cout << ''In base class f2'' << endl; }
};

class derivedClass
{
  public:
    derivedClass()
      { cout << ''In derived class constructor'' << endl; }
   ~derivedClass()
      { cout << ''In derived class destructor'' << endl; }
    void f1() { cout << ''In derived class f1'' << endl; }
    void f2() { cout << ''In derived class f2'' << endl; }
};
```

What is the ouput of the following program segment? Please write your answers next (or below) to the functions called in main.

```
main()
{
  baseClass x;


  derivedClass y;


  baseClass *z = new derivedClass;


  x.f1();


  x.f2();


  y.f1();


  y.f2();
```

```
    z->f1();

    z->f2();

}
```

4. A virtual destructor similar to a virtual function, except that instead of any function, the destructor is made virtual. The behavior of a virtual destructor is similar to that of any virtual function. Show with an example how a virtual destructor can be implemented. Why would anyone need to implement a virtual destructor? Does it serve any purpose to have a virtual destructor, as opposed to having a regular, non-virtual destructor? . . . (3 marks)

5. Multiple inheritance is a case where one class is derived more two or more classes. For example, the following declaration shows that square is derived from both shape and polygon.

```
class square : public shape, polygon
```

   What are the advantages of multiple inheritance? What are the negative points and potential problems of multiple inheritance? Support your arguments with examples for both advantages and disadvantages. . . . (3 marks).