

W3101 Programming Languages – C++ Midterm Oct 08, 2007

Name:

Student Id:

1. Any program that can be written in C++ can be done in C. Do you agree with the statement? If so, why do we need C++ and what do we gain by using C++ instead of C? If not, describe one program (at a high level) that we can do only in C++ and not in C. Be brief and precise. Explain your ideas clearly. ... (3 marks).

Answer: The answer is true. Any program that can be written in C++ can be written in C, though it may be cumbersome to implement OOP features like data encapsulation, inheritance, etc. Syntactically, C++ offers a lot features (like class definition, inheritance, etc.). But them can be emulated and implemented in C also.

2. Write the constructor code for the executive class given below that takes an integer value called “amount” as input and sets it as the executive’s salary. ... (2 marks).

```
class employee
{
    private:
        int salary;
    public:
        employee(int x) : salary(x) { }
};

class executive : public employee
{
    // Write the constructor code here
    executive (int amount) : employee (amount) { }
    // The above line shows the constructor code
}
```

3. What is the output of the following program segment? Please write your answers next (or below) to the functions called in main. ... (4 marks).

```
class baseClass
{
    public:
        baseClass() { };
```

```

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

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

main()
{
    baseClass x;
    derivedClass y;
    baseClass *z = new derivedClass;

    x.f1(); // Answer: In base class f1

    x.f2(); // Answer: In base class f2

    y.f1(); // Answer: In derived class f1

    y.f2(); // Answer: In derived class f2

    z->f1(); // Answer: In derived class f1

    z->f2(); // Answer: In base class f2

}

```

```

4. class employee
    {
        private:
            int salary;
        public:
            employee(..) { /* code */ }
    }

```

What are the different ways in which any external (not a member of this class) function or class access the private member of the class employee shown above? You are free to add member functions to the class, if you wish. Write one line code (to show how it is done) for each way you list in your answer. ... (3 marks).

Answer: Any external class or function can access the private members using

- (a) Public access methods — e.g., <type> getMember() or void setMember(<type> data)
- (b) friend function — e.g., friend void myFriendFunction (<type> args)
- (c) friend class myFriendClass;

5. Answer the questions given below. ... (3 marks).

a) What is an abstract class? What is its use?

Answer: An abstract class is a class that has at least one pure virtual function, i.e., the class has at least one virtual function that is equal to zero, without any body (implementation) for the function. Abstract class can have data members and other virtual and non-virtual functions that are not pure. They define the class and its members without necessarily giving the implementation details. Classes derived from the abstract class need to implement the pure virtual functions and other class specific functions. Objects of abstract classes cannot be instantiated.

b) How can you make the employee class given below an abstract class?

```
class employee
{
    private:
        int salary;
        int ssn;
    public:
        employee() { ssn = -1; salary = -1; }
        int getSSN { return ssn; }
        int getSalary();
        void
}
```

Answer: Make one function a pure virtual function, e.g.,
virtual int getSalary () = 0;

c) What is the outcome of the code given below?

```
main()
{
    employee x;
    cout << ``SSN: `` << x.getSSN() << endl;
}
```

Answer: This results in a compiling error; objects of abstract classes cannot be instantiated.