University of Technology

Department of Computer Sciences

Final Examination 2011-2012

Subject: Object oriented programming *year: 2012*

Division: *Time: 3 hours*

Examiner: Dr. Rehab F. hassan *Date:*

Answer 5 Question Only, 10 marks for each question

Q1: Declare a class named Triple with three private data members (floats) x, y, and z. Provide public functions for setting and getting values of all the private data members. Define a constructor that initializes the values to user-specified values or, by default, sets the values all equal to 0. Also overload the following operators:

- Addition so that corresponding elements are added together
- Output so that it displays the Triple in the form "The triple is (x, y, z)."
- Assignment that copies x to z, y to x, and z to y.
- Increment so that x and z are increased by one each.

```
Class Triple
{private:
        float x, y, z;
 public:

        Triple (float a =0, float b=0 , float c=0)
                { x=a;  y=b;  z=c;}
        void SetX   (float a)   { x=a; }
        void  SetY   (float a)  { y=a; }
        void  SetZ   (float a)   {z=a; }
        float GetX (void)       {return (x); }
        float GetX (void)       {return (x); }
        float GetX (void)       {return (x); }
        Triple operator +  (Triple op)
                { Triple res;
                  Res.x = x + op.x;
                  Res.y = y + op.y;
                  Res.z = z + op.z;
                  Resturn res;          }
        void  operator & (void)
                { cout << "The triple is ("<<x<<", "<<y<<" "<<z<<").";  }
        void operator = (Triple op)
                { x=op.z;
                  y=op.x;
                   z=op.y;    }
        void operator ++ (void)
                {   x++;   z++;   }
};
```

Q2: Assume that registration codes, each consisting of a sequence of characters, are implemented using the constants and class declaration below.

```
const int MINLEN = 5;
const int MAXLEN = 15;
class Code
{ public:
        Code( );                        // length is 0, capacity is MAXLEN
        Code(const char *s);              // code has chars in s, length is s.length()
        bool IsValid( );                  // returns true if code is valid, else false
        int Length( );
```

```
            // returns # characters in the code
private:
        int myLength;
        char myChars[MAXLEN];
};
```

Write the complete definition of the two constructor functions, and the length function. Write the member function IsValid, which returns true if a code is valid and returns false otherwise. A code is valid if it satisfies the following criteria.
The length is greater than or equal to MINLEN and less than or equal to MAXLEN - 2,
i.e., MINLEN <= Length() <= MAXLEN - 2
Each character is either a digit ('0'..'9') or a capital letter ('A'..'Z').

```
const int MINLEN = 5;
const int MAXLEN = 15;
class Code
{ public:
        Code( );                        // length is 0, capacity is MAXLEN
        Code(const char *s);             // code has chars in s, length is s.length()
        bool IsValid( );                 // returns true if code is valid, else false
        int Length( );                   // returns # characters in the code
private:
int myLength;
char myChars[MAXLEN];
};
        Code :: Code (void) {mylength=0; strcpy(myChars,"");  }
        Code :: Code (const char *s) {mylength=strlen(s); strcpy(myChars,s);    }
        Code :: int Code (void)
                {for (myLength i=0 ; myChars[myLength]!='\0' ; ++myLength)  ;  }
        Code :: bool IsValid(void)
                {if ((myLength <MINLEN) || (myLength>MAXLEN-2 ))
                                return (false);
                  for (int i=0 ; i<myLength ; ++i)
                   if  ! (myChars[i]>='0' && myChars[i]<='9')&&
!(myChars[i]>='A' && myChars[i]<='Z'
    return (false);
                return (true);      }
```

Q3: Answer by true or false for the following:

1. Pointers to a base class may be assigned the address of a derived class object. **true**
2. A virtual method must be overridden in a derived class. **true**
3. If a binary operator is overloaded using a global function, then two parameters are required. **true**
4. Destructor function may be called before constructor function **false**
5. Friend function can access the public member only
6. Constructor function could be overloaded **true**
7. Classes may contain variables of different data types and functions **true**
8. A base-class initialize must always be provided in the derived-class constructor to call the base-class constructor. **false**
9. A subclass class cannot have a method with the same name as a base class method. **false**
10. Constructor function may return value be return statement. **false**

Q4: Answer the following :

1. Explain how "multiple" inheritance differs from "single" inheritance.

   By multiple inheritances we mean the subclass inherit its properties and method from two or more base class, while in single inheritance the subclass inherit from one base class.

2. Give the declaration of a class D that publicly inherits from class B, and privately inherits from class C

   Class D: public B, private C

3. Write one or more C++ statements that dynamically allocates an array of 42 integers and initializes each one to zero.

```
int  * a = new (42 [int]);
for (int i=0 ; i<42 ; ++i)   *(a+i) = 0;
```

4. Give three properties for the following: friend function, Destructor function

   - Friend function should be global function.
   - Can access all the members of the class
   - Its parameters are usually one or more of class's objects

   - Destructor function has no return value
   - Has no parameter
   - Called automatically when the object is finished

5. Give two situations where we need to use the scope operator.

   - When the member functions are defined outside the class scope, example:

     Point:: void Printing (void) { .....}

   - When the parameters of a member function have the same name of the private members, example:

```
class Point
{private:
        int x , y;
 public:
        Point(int x , int y)
        { Point::x = x ; Point::y = y; }
```

Q5: Show the output of the following program:

a.

```
#include<iostream>
class A{
public:
   int f( ) {return 1;}
   virtual int g( ) {return 2;}
};
class B: public A{
public:
   int f( ) {return 3;}
   virtual int g( ) {return 4;}
};
class C: public A{
public:
   virtual int g( ) {return 5;}
};
int main( ){
A *pa;
A a;
B b;
C c;
pa=&a; cout<<pa -> f( )<<endl;
cout<<pa -> g( )<<endl;
pa=&b; cout<<pa -> f( ) + pa ->g()<<endl;
pa=&c; cout<<pa -> f( )<<endl;
cout<<pa -> g( )<<endl;
return 0;
}
```

1 2 5 1 5

b.

```
class exforsys
{
public:
exforsys(void) { x=0; }
void f(int n1)  {  x= n1*5;  }
void output(void) { cout << "n" << "x=" << x; }
private:
    int x;
};
class sample: public exforsys
{
public:
sample(void) { s1=0; }
void f1(int n1)  {  s1=n1*10;  }
void output(void)
{
exforsys::output( );
cout << "n" << "s1=" << s1;
}
private:
    int s1;
};
int main(void)
{
sample s;
s.f(10);
s.f1(20);
s.output( );
}
```

nx=50ns1=200

Q6: Choose the correct answer:

1. If we declare a queue template class with dynamic allocation, then we can declare queue of integer of length 100 as:

A. int queue Q1[100]; B. template queue <int> Q1(100);
C. queue <int> Q1(100); D. class int queue (100) Q1;

2. Occasionally we may need to grant a function access to the non public members of a class. Such an access is obtained by declaring the function as:

A. A **friend of the class.** B. A protected part of the class.
C. A constructor function. D. A subclass member function.

3. When a class member function is called, it receives an implicit argument which denotes the particular object (of the class) for which the function is invoked. Within the body of the member function, one can refer to this implicit argument explicitly as:

A. Structure **B. This** C. Pointer D. - >

4. Operator overloading means:

A. defining additional meanings for the predefined operators. B. defining inline functions.
C. defining additional arithmetic operators. D. defining additional logical operators.

5. Which statement about operator overloading is false?

A. New operators can never be created.
B. **Certain overloaded operators can change the number of arguments they take.**
C. The precedence of an operator cannot be changed by overloading.
D. Overloading cannot change how an operator works on built-in types.

6. When class B is inherited from class A, what is the order in which the constructers of those classes are called

A**. Class A first Class B next** B. Class B first Class A next
C. Class B's only as it is the child class D. Class A's only as it is the parent class

7. Providing access to an object only through its member functions, while keeping the details private is called.

A. Information hiding**.** **B. Encapsulation.** C. Inheritance D. Abstract data type

8. A member function

A. is always public B. is always private **C. can be public or private** D. cannot be defined.

9. Can two classes contain member functions with the same name?

A. No. B . Yes, but only if the main program does not declare both kinds
B. Yes, but only if the two classes have the same name. **D. Yes, this is always allowed**.

10. To define a generalized sorting algorithm, we should use one of the following techniques:

A. Function Overloading. **B. Templates** C. Polymorphism. D. Inheritance.