Jordan University of Science and Technology
CS112 Introduction to programming Language
First Semester 2011-2012
First Exam- Form (D)

Name: **Id:**
Section: **Instructor:**

Question 1: Write a code for the following UML class Rectangle (40 points)

Rectangle
- width: double **- length: double**
+ Rectangle(int =1, int =1) **+ set(int, int): void** **+ Area(): double** **+ Addition(const Rectangle &) : Rectangle** **+ ~Rectangle()**

Write a code for the header of Rectangle class (Class definition with the member variables and the memeber functions **prototypes**) (10 points)

```
Class Rectangle          // 1 point
{
        private:    // or by default private    (1 point)
                double length;   // 1 point
                double width;  // 1 point
        public:    // 1 point
                void set(int, int );    // 1 point
                Rectangle Addition (const Rectangle &);   //1 point
                Double Area( );       //1 point
                Rectangle (int =1, int =1);    // 1 point
                ~Rectangle ( );   // 1 point
};


-1 point for each of the following
if there is No { };
if there is No ":"
if there is No "="
```

Wrtie the implementation of **the Constructor** function that will take two default parameters and set the value of the length and the width. (6 points)

```
Rectangle : : Rectangle (int a, int b)          //  3 points
              ( -1 if the student wrote Rectangle : : Rectangle (int a=1, int b=1))
              (-1 if the student wrote void Rectangle : : Rectangle (int a, int b) )
              (-1 if the student wrote Rectangle (int a, int b) )
              (-2 if the student wrote Rectangle : : void Rectangle (int a, int b) )
             (-2 if the student wrote the Word "Constructor" instead of "Rectangle")
{
   length= a;   // or width =a;      // 1.5
   width=b;    // or length=b;      //1.5
                            (-1 for length=1    -1 for width=1)
                            (-1    if a= length       and         b= width)
}
```

Wrtie the implementation of **the Destructor** function that will print the value of the length and the width of the rectangle. (6 points)

```
Rectangle : : ~Rectangle ()          //  3 points
              ( -1 if the student wrote     ~Rectangle : : Rectangle ()  )
              (-1 if the student wrote     void Rectangle : : ~Rectangle () )
              (-1 if the student wrote     ~Rectangle () )
              (-2 if the student wrote    Rectangle : : void ~Rectangle ( ) )
             (-2 if the student wrote  the worde "Destructor" instead of "Rectangle")
{
         cout<<length <<"   "<<width<< endl;            // 3 points
         // -2   (if the cout<< RectObject . length<<"    "<<RectObject . width<<endl;
}
```

Wrtie the implementation of **set** function that takes two parameters and gives them to length and width memebr variables. (5 points)

```
Void Rectangle : : set(int  a  , int  b )        //    2   points
                                  (-1 if the student wrote  void set(int a, int b)   )
                                  (-2 if the student wrote Rectangle:: void set(int a , int b) )
{

length =a;                  // 1.5   points                  (-1  if the student wrote a= length)
width =b;                   // 1.5    points                 (-1 if the student wrote b=width)
//o r   width =a;  length =b;
}
```

Wrtie the implementation of **Addition** function that takes one object parameter and returns an object. The returned object is the summation of the length and the width of the calling object and the parameter object. (8 points)

```
Rectangle Rectangle:: Addition ( const Rectangle & obj)      // 2 points
                (-1  if the student wrote Rectangle Addition (const Rectangle & obj)
                (-2  if the student add two parameters  Addition( Rectangle obj1, Rectangle obj2)

{
        Rectangle  Temp;                    //   1  point
        Temp . length =   length  +   obj . length ;        //  2 points
                        ( -1   if the student wrote   Temp . length=  Rect1. length + Rect2.length;)

        Temp . width =  width  +  obj. Width ;        // 2 points
                        ( -1   if the student wrote   Temp . width=  Rect1. width + Rect2.width;)


        return Temp;                        //1 point
}
```

Wrtie the implementation of **Area** function that returns the area of the calling object.
(Note: Area =length * width) (5 points)

```
double  Rectangle : : Area( )      //   2 points
                (-1  if the student wrote double Area() )
                (-2 if the student wrote Rectangle : :  double Area ( ) )

{
        return length  *  width;            //  3  points
                // (- 2 if the student didn't return value )
                // (-2 if the student wrote return Rect. Length *  Rect . width

}
```

If the main code is as the following**: (Don't write the main function)**

```
int main()
{

       Rectangle Rec1, Rec2(10, 10), Rec3;
       Rec1.set(5,8);
       cout<< Rec1.Area()<<endl;
       cout<< Rec3.Area()<<endl;
       Rec3=Rec1.Addition(Rec2);
       return 0;
}
```

Question 2: Find five (5) Errors in the following code and explain why it is an error (10 Points)

```
1-#include<iostream>
2-#include<string>
3-using namespace std;
4-struct Student
5-{
6-     double GPA;
7-     char status='P';
8-      string FName;
9-};
10-void main()
11-{
12-    FName="Ali";
13-    Student s1,s2;
14-    cin>>s2;
15-    s1::GPA=98.5;
16-    s1=s1+s2;
17-}
```

1- status= 'p'	" illegal ="
2- Fname="Ali"	" should be obj.Fname="Ali" "
3- cin>>S2	"illegal cin>>objects "
4- s1::GPA	"should be s1.GPA"
5- s1=s1+s2	"ilegal + "

Question 3: What is the output of the following code? (50 Points)

A

```
#include<iostream>
using namespace std;
void main()
{
  enum Shape{square ,rectangle ,circle ,traingle};
  Shape S1 , S2 ;
  S1 = rectangle ;
  S2 = static_cast<Shape>(S1+4);
  cout<<S1<<" "<<S2<<endl;
}
```

1 5
(4 points)

B

```
#include<iostream>
#include<string>
using namespace std;

void main()
{
 string Salut = "Good Morning Jordan";
 string Greetings= Salut.substr(0,13)+"Students";
 cout<<Salut<<endl;
 cout<<Greetings.length()<<endl;
}
```

Good Morning Jordan
(2 points)
21
(2 points)

C

```
#include<iostream>
using namespace std;
enum GPA {A,B=4,C,D=5,E};
void main()
{
      GPA X1,X2;
      X1=C;
      X2=D;
      cout<<"X1:"<<X1<<", X2:"<<X2<<endl;
}
```

X1:5, X2:5 //3 points

D

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string S="Good Morning Students Do your best";
	int count=0;
	int i=0;
	while(S.find(" ", i)!=string::npos)
	{
		i=S.find(" ",i);
		i++;
		count++;
	}
	cout<<count<<endl;

	return 0;
}
```

5 (2 poinst)

E

```
#include<iostream>
#include<string>
using namespace std;

struct A
{
	int var1;
	int Arr[3];
};
struct B
{
	int var1;
	A varSt;
};
void main()
{
	A objA;
	B objB;
	objA.var1=5;
	for(int i=0; i<3; i++)
		objA.Arr[i]=i+10;
	objB.var1=20;
	objB.varSt=objA;
	objB.varSt.var1++;
	cout<<objB.var1<<"   "<<objB.varSt.var1<<endl;
	for(int i=0; i<3; i++)
		cout<<objB.varSt.Arr[i]<<"  ";
	cout<<endl;
}
```

20 6
10 11 12

(10 points)

F

```
#include<iostream>
using namespace std;

class A
{
	int x;
	int y;
public:
	A(){ x = 0 ; y = 0 ; }
	A(int a,int b=0) {x=a ; y=b ; }
	void f1(int a) { x = a ; y = a*2 ; }
	void print()const{ cout<<x<<" "<<y<<endl;}
	~A(){ cout<< x << " " << y <<endl;}
};
void main()
{
	A obj1,obj2(3,5),obj3(10);
	obj1.f1(1);
	cout<<"obj1 : ";
	obj1.print();
}
```

obj1 : 1 2 **3 points**
10 0 **2 points**
3 5 **2 points**
1 2 **2 points**

G

```
#include<iostream>
#include<string>
using namespace std;

class A
{
	int x, y;
public:
	A(){x=1; y=2; cout<<"First "<<endl;}
	A(int a, int b){x=a; y=b; cout<<"Second "<<endl;}
	~A(){cout<<x<<"  "<<y<<"  Destructor"<<endl;}
	void increment(){x++; y++;}
};

int main()
{
	A obj1, obj2(9,6);
	{
		A obj3(4,7);
		obj1.increment();
	}
	A obj4(5,7);

	return 0;
}
```

First 1 poin
Second 1 point
Second 1 point
4 7 Destructor 2
Second 1
5 7 Destructor 2
9 6 Destructor 2
2 3 Destructor 2

The same as Form A

H

```
#include<iostream>
#include<string>
#include<set>
#include<cctype>
using namespace std;

class A
{
	static int x;
	int y;
public:
	void fun1();
	static void fun2();
	A();
};
int A::x=5;

A::A()
{
	x=5; y=50;
}
void A::fun1()
{
	x=x+2;
	y=y+2;
	cout<<x<<"   "<<y<<endl;
}

void A::fun2()
{
	x=x+5;
	cout<<x<<endl;
}

int main ()
{
	A::fun2();
	A obj1, obj2, obj3;
	obj1.fun1();
	obj2.fun1();
	obj3.fun2();
	return 0;
}
```

10
7 52
9 52
14

6 points