MILITARY COLLEGE OF SIGNALS
MID TERM EXAM - BESE15
OBJECT ORIENTED PROGRAMMING PARADIGM

Instructor: Aisha Khalid Khan

Time: 90 Minutes
Max Marks: 30

Instructions

1. Understanding of question is part of Examination.
2. State any assumptions, in the beginning of your solution, which you are making about the question.
3. Assume in all programs necessary header files are included.
4. Answer all questions on the answer booklet and do not expect that anything written on the question paper will be considered for marking.
5. Do not use a lead pencil for answering any of the questions
6. Attach the question paper at the back of your answer booklet.
7. This question paper has **4** pages.

Question 1: Suppose your program contains the following class definition: **[3]**

```
class Automobile
{
public:
      void setPrice(double newPrice);
      void setProfit(double newProfit);
      double getPrice();
private:
      double price;
      double profit;
      double getProfit();
};
```

and suppose that all of the class methods have been defined. The **main** function of your program contains the following declaration

```
Automobile hyundai, jaguar;
```

Which of the following statements are allowed in the **main** function of your program? For illegal statements, explain in one sentence why they are illegal?

```
 i.   hyundai.price = 4999.99;
ii.   jaguar.setPrice(30000.97);
iii.  double aPrice, aProfit;
iv.   aPrice = jaguar.getPrice();
 v.   aProfit = jaguar.getProfit();
vi.   aProfit = hyundai.getProfit();
```

Question 2: Given the definitions **[3]**

```
const int x=17;
class A
{
public:
      A();
      A(int n);
      int f() const;
      int g(const A& x);
private:
      int i;
};
```

What is the consequence of using the **const** keyword at 3 places in the above code segment? What guarantee does the compiler provide in each of the three cases?

Question 3: Find compile-time errors in the following code segments, and explain why these errors occur. What can you do to correct them? The line numbers on the left are not part of the code. Refer to these line numbers when answering the questions. **[2.5+2.5]**

(a)

```
1. class stack
2. {
3.    public:
4.    void push(const int data) { arr[top++] = data;}
5.      int pop() const { return arr[--top];}
6.   protected:
7.    int arr[100];
8.   private:
9.    int top = 0;
10. };
```

(b)

```
1. class Base{
2. public:
3.    void init() { count = sum = num =0; }
4. protected:
5.    int count;
6. private:
7.    int sum;
8.    const int num;
9. };

10. class Derived: public Base
11. {
12.    public:
13.         void init() { avg = 0;}
14.      int getSum()  { return sum;}
15.   private:
16.         int avg;
17.   };
```

Question 4: What would be the output of the following program segments? **[1.5 +2.5]**

(a)

```
class ss
{
      static int c;
      public:
            static void set() { c++; }
            void display(){cout<< c; }
};
int ss::c=12;

void main()
{
ss obj;
obj.set();
ss::set();
obj.display();
}
```

(b)

```
class C {
      private:
            static int n;
      public:
            C(){ cout << "\nC's constructor # " << ++n ; }
            ~C(){ cout << "\nC's destructor # " << n-- ; }
};
int C::n=0;
class A {
      protected:
            C c,d;
      public:
            A(){ cout << "\nA's constructor"; }
            ~A(){ cout << "\nA's destructor"; }
};
class D : public A {
public:
      D(){ cout << "\nD's constructor"; }
      ~D(){ cout << "\nD's destructor"; }
};

void main(){
D a[2];
cout<<"Destruction order"<<endl;
}
```

Question 5: Consider the following program that implements single as well as multiple inheritance. Some statements, in **main** are incorrect. List them and explain the reason of errors. How can these errors be corrected? **[5]**

```
class A{
      int i1;
 protected:
      int i2;
 public:
   void seti(int inp){ i1=inp; i2=0; }
   void f1(char c[])
      { cout << c << endl;}
};
class B:public A{
      int i1;
 public:
  int i3;
  void seti(int inp){  i1=inp; i3=3; }
  void f1(int i){ cout << i << endl;}
};
class C:private A{
      int i1;
};
class D{
 public:
  void f1(char c[])
    {cout << c << endl;}
};
class E:public B,public D{
      int i1;
};
```

```
void main()
{
 A a;
 B b;
 C c;
 E e;
 a.i2=1;
 b.A::seti(3);
 a.seti(2);
 b.f1("INPUT1");
 c.seti(4);
 c.f1("INPUT2");
 e.seti(5);
 e.B::i3=7;
 e.f1("INPUT3");
}
```

Question 6: **[10]**

For the following questions, use **const** where appropriate. You do not have to write the **main** program.
Consider a class Time which contains two private data members **hour** and **minute** of type **int**.

(a) Implement the class. The class's public interface contains:
- A 2-argument constructor to initialize hour and minute to values provided in the main program. If no values are provided in the main program, the default values assigned to hour and minute should be zero
- A **display()** function to show time in the **hour:minute** format. (See **main** program below)
- A **getHour()** function which returns hour
- A **getMinute()** function which returns minute

(b) Write a non-member (global/free-standing) function to overload the postfix increment operator. When this operator is called on a time object, it increments the value of minute by 1. If minute exceeds 59, then minute is set to 0 and hour is incremented by 1.

(c) Write a global function swap which takes two Time values as arguments and swaps their members.

A sample main program and its intended output is given below:

```
void main()
{
      Time t1(1,59);
      Time t2(8,40);
      t1++;
      t1.display(); //prints "time is: 2:0"
      swap(t1,t2);
      t1.display(); //prints "time is: 8:40"
      t2.display(); //prints "time is: 2:0"
}
```

(d) Using public inheritance, derive a class **SecTime** from **Time** which adds its own private data member **second** of type **int**. Implement the functions of class **SecTime** so that the user can interact with the program in the following way.

```
void main()
{
      secTime s1(10,20);
      s1.display(); //prints "time is: 10:20:0"
      secTime s2(1,20,45);
      s2.display(); //prints "time is: 1:20:45"
      secTime s3;
      s3 = s1 + s2; // adds s1 to s2 and returns the sum which is assigned to s3. For this function, don't bother
      //about setting minute/sec to 0 if they exceed 59
      s3.display(): //prints "time is: 11:40:45"
}
```

x----------x----------x----------Good Luck ☺----------x----------x----------x