KING SAUD UNIVERSITY COLLEGE OF COMPUTER AND INFORMATION SCIENCES COMPUTER SCIENCE DEPARTMENT			
CSC 113: Computer Programming II	**Midterm 1 (Duration: 1:30)**	**2nd Semester 1438-1439**	
Student Name (Arabic)	**Student ID**	**Section Number**	**Serial Number**

Question#1: Multiple Choice Questions *(5 pts.)* *(0.5 for each)*

For each statement there is a list of options. Choose the option that would be the valid one.

1	2	3	4	5	6	7	8	9	10
c	**a**	**b or c**	**a**	**b**	**a**	**b**	**a**	**b**	**a**

1- What is the output of following Java program?

```
class Parent {
    public void function() {
System.out.println("Super Class");
  }
}
class Child extends Parent {
    private void function() {
System.out.println("Sub Class");
  }
}
public class App1 {
 public static void main(String
args[]) {
        Parent p = new Child();
        p.function();
    }
}
```

a. Sub Class
b. Super Class
c. Compiler Error
d. Sub Class
   Super Class

2- What is the output of following Java program?

```
class Base {
   int i = 10;
}
class sub extends Base {
   int i = 20;
}
public class App2
{
   public static void main(String[]
args){
       Base b = new sub();
       System.out.println(b.i);
   }
}
```

a. 10
b. 20
c. Compiler Error
d. 10
   20

3- An inherited protected attribute becomes ……. in the subclass

a) public member
b) private member
c) protected member
d) static member

4- If super class and sub class have same variable name, which keyword should be used to use super class?

a) super
b) this
c) Name of sub class
d) Name of super class

5- What is the output of following Java program?

```
class G {
    String s = "Grand Parent";
}
class P extends G {
    String s = "Parent";
    P(){
System.out.println(super.s);}
}
class C extends P {
    String s = "Child";
    C(){ System.out.println(s);}
}
public class App3 {
    public static void
main(String[] args){
       C child = new C();
      System.out.println(child.s);
    }
}
```

a. Grand Parent
Child

b. Grand Parent
Child
Child

c. Grand Parent
Parent
Child

d. Child
Child

6- What is the output of following Java program?

```
class A{
    int i;
    public A(int i) {
        this.i = i--;
    }
}

class B extends A{
    public B(int i) {
        super(++i);
        System.out.println(i);
    }
}

public class App4 {
    public static void main(String[] args)
{
        B b = new B(6);
    }
}
```

a. 7
b. 6
c. 5
d. Compiler Error

7- Constructors are inherited to sub classes.

a. True.

b. False.

8- Can a class be extended by more than one classes?

a. Yes.

b. No.

9- Once a subclass is formed, no further inheritance from that subclass is allowed.

a. True.

b. False.

10- A subclass can effect state changes in superclass private members only through public, protected methods provided in the superclass and inherited into the subclass.

a. True.

b. False.

Question#2: The following code will generate compiler error. Find out and correct the error in the below codes. ***(3 pts.)*** ***(0.5 for each-either for correction or explanation)***

1.
```
class A {
   public A(int x) {
     System.out.println(x);
   }
}
class B extends A {
   public B(){
      System.out.println(2);
   }
}
```

Error Correction:

No Default Constructor in Class A.

Or

No explicit call to super(int)

2.
```
class A {}
class B extends A{
   public B(){
    System.out.println("Start of B");
    super();
    System.out.println("End of B");
   }
}
```

Error Correction:

A call to super must be first statement in constructor.

3.
```
class A {}
class B extends A { }
class C extends A { }
class MainClass {
   public static void main
(String[] args) {
      B b = new B();
      A a = b;
      C c = b;
   }
}
```

Error Correction:

Incompatible types. b can't be assign to c (they are not related by inheritance)

4.
```
class A{ }
class B extends A {
   public void MyMethod(){}
}
class MainClass{
   public static void main (String[]
args) {
      A b = new B();
      b.MyMethod();
   }
}
```

Error Correction:

`MyMethod` is not available in class A. Casting is needed.

```
((B)b).MyMethod();
```

5.

```
class A {
 public final int calculate(int a,
int b)
  { return a+b; }
}
class B extends A {
 public int calculate(int a, int b)
{
      return a*b; }
}
class MainClass {
  public static void main(String
args[]){
      B b = new B();
      System.out.print("b is " +
b.calculate(0, 1));
   }
}
```

Error Correction:

Method calculate is final and can't be overridden.

6.

```
class A {
   public int MyMethod(){ return 0;}
}
class B extends A {
   public void MyMethod(){}
}
class MainClass{
   public static void main(String[]
args) {
      B b = new b();
      b.MyMethod();
   }
}
```

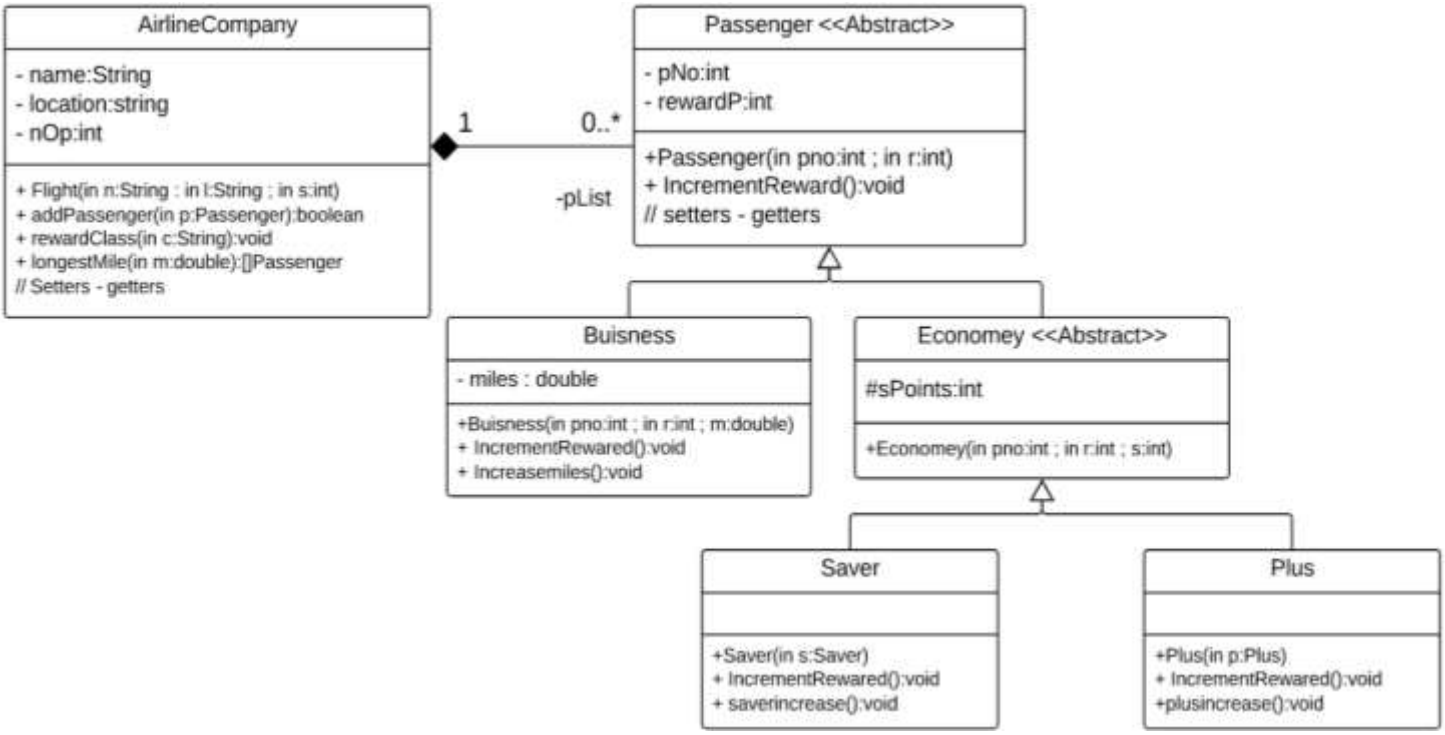
Error Correction:

Incorrect method signature in class B , incorrect overridden.

OR

```
B b = new b();
```

Question#3: Consider the following UML Diagram: *(7 pts.)*



The descriptions of UML class:

Passenger class***:***

- Attributes:
  - ***pNo***: the passport number of the passenger.
  - ***rewardP:*** passenger reward points.
- Methods:
  - ***Passenger(pNo:int; r: double*)*:** Constructor.
  - ***IncrementRewared(): void:*** An *abstract* method to increment the passenger reward points.

Business class***:***

- Attributes:
  - ***miles***: the number of miles travelled by the business passenger.
- Methods:
  - ***Business (pNo:int; r:double; m:double)***: Constructor.
  - ***IncrementRewared():void:*** A method to increment the passenger reward points as follow:
    - ***For every 100 miles:*** Increase 10 points.

Economy class

- Attributes:
  - ***sPoints:*** Saving points.
- Methods:
  - ***Economy (pNo:int; r:int; s:int)***: Constructor.

Saver class***:***

- Methods:
- ***Saver (s:Saver)***: Constructor.
- ***IncrementRewared():void*** A method to increment the passenger reward points as follow:
  - ***For every 10 saving points:*** increase 10 points.

Plus class***:***

- Methods:
- ***Plus (p:Plus)***: Constructor.
  - ***IncrementRewared():void*** A method to increment the passenger reward points as follow:
    - ***For every 10 saving points:*** increase 12 points.

AirlineCompany class***:***

- Attributes:
  - ***name:*** Airline Name.
  - ***location***: the location of the airline company.
  - ***nop:*** Number of passengers in the airline.
- Methods:
  - ***AirlineCompany(n: String; l:String; size:int):*** Constructor to initialize all attributes.
  - ***addPassenger(p:Passenger):boolean:*** this method adds passenger to the Airline company *if possible*. <u>There is a maximum of only 50 business passengers on the company</u>. If adding a passenger is not possible, the method will display an appropriate message and return false.
  - ***rewardClass(c:String): void:*** this method reward all passengers of the received class by incrementing the *rewardpoints.*
    - *Note*: the received class can be <u>either Business or Economy</u>.
  - ***longestMile(m:double):[] Passenger:*** this method returns an array *contains only* all **Business** passengers who have <u>miles</u> grater than or equal to **m**.
    - *Hint:* Be sure to use DEEP COPY where appropriate.

Translate into Java code the following selected methods from class ***AirlineCompany***

a)
```
public boolean addPassenger(Passenger p) // Total: 2.75pts

if (nop==plist.length)  { //0.25pt
          System.out.println("Sorry, the list of passengers is
full");
          return false;
       }
       if (p instanceof Saver) //0.25pt
          plist[nop++] = new Saver((Saver)p); //0.25pt for
correct instantiation with casting (No casting 0)
       else if (p instanceof Plus) //0.25pt
          plist[nop++] = new Plus((Plus)p); //0.25pt for
correct instantiation with casting (No casting 0)

       else if (p instanceof Business) { //0.25pt
          // counts number of business passengers in plist
          int Bcnt=0;
          for(int i=0; i<nop; i++) //0.25pt
             if (plist[i] instanceof Business) //0.25pt
                Bcnt++; //0.25pt

          if(Bcnt>=50) { // or Bcnt==50 //0.25pt
             System.out.println("There is a maximum of only 50
business passenger");
             return false;
          }
          else
             plist[nop++] = new
Business(p.getPNO(),p.getP(),((Business)p).getmiles());//0.25p
t for for correct instantiation wit casting (No casting 0)
       } // business case
       return true;
```

Grading Details for method ***addPassenger***:

- Check array (Plist) is not full ***0.25pt***
- Check Saver case (instance of Saver) ***0.25pt***
  - Add saver object correctly with casting ***0.25pt***
- Check Plus case (instance of Plus) ***0.25pt***
  - Add plus object correctly with casting ***0.25pt***
- Check Business case (instance of Business) ***0.25pt***
  - Check num of business passengers doesn't exceeds the maximum ***1pt***
    - (***.25*** loop + ***.25*** checking if array element is instance of business + ***.25*** incrementing counter + ***.25*** if>=50

- Add business object correctly with casting ***0.25pt***

b)
```
public void rewardClass(String c) // Total: 2.25pts
      if (c.equals("Economy")) { //0.25pt
         for (int i=0; i<nop;i++) //0.25pt
            if(plist[i] instanceof Saver || plist[i]
instanceof Plus) //0.5pt
               plist[i].IncrementRewared();//0.25pt
      }
      else { // c.equals("Business") //0.25pt
         for (int i=0; i<nop;i++) //0.25pt
            if(plist[i].getClass().getName().equals(c)) // or
if(plist[i] instanceof Business  //0.25pt
               plist[i].IncrementRewared();//0.25pt
      }
```

OR

//student may consider that the received "Economy" is either "Plus" or "Saver" which makes it correct

```
for (int i=0; i<nop;i++) //0.25pt
            if(plist[i].getClass().getName().equals(c))
//1.75pt
               plist[i].IncrementRewared();//0.25pt
```

Grading Details for method ***rewardClass***:

- Check if c is Economy ***0.25pt***
  - Loop over the filled elements ***0.25pt***
    - Check if array element is instance if saver or plus ***0.5pt***
    - IncrementRewared ***0.25pt***

- Check if c is Business ***0.25pt***
  - Loop over the filled elements ***0.25pt***
    - Check if array element is instance of business ***0.25pt***
    - IncrementRewared ***0.25pt***

c)
```
public Passenger[] longestMile(double m) // Total: 2pt
      int pCnt=0;
      for (int i=0;i<nop;i++)
         if(plist[i] instanceof Business &&
((Business)plist[i]).getmiles()>=m)
            pCnt++;
      Passenger[] longestp = new Passenger[pCnt]; //0.5pt

OR

int pCnt=0;
      for (int i=0;i<nop;i++)
         if(plist[i] instanceof Business )

            pCnt++;
      Passenger[] longestp = new Passenger[pCnt]; //0.5pt

OR       Passenger[] longestp = new Passenger[nOp]; //0.5pt

      int j=0;
      for (int i=0;i<nop;i++) //0.25pt
         if(plist[i] instanceof Business &&
((Business)plist[i]).getmiles()>=m) //0.5pt (0.25 for instance
of + 0.25 for casting)
            longestp[j++]=new
Business(plist[i].getPNO(),plist[i].getP(),((Business)plist[i]
).getmiles());//0.5pt (0.25 for using separate counter + 0.25
for new  )
      return longestp; //0.25pt
```

Grading Details for method ***longestMile***:

- Count num of business passengers that satisfy condition ***0.25pt***
- Create array of passengers correctly ***0.25pt***
- Loop over the filled elements ***0.25pt***
    - Check if the element is instance of Business & satisfy condition ***0.5pt***
    - Add to the array a new business instance correctly ***0.5pt***
- return the array ***0.25pt***