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

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	~~b~~	c	a	~~d~~	~~b~~	b	b	b	a
✓	✗	✓	✓	✗	a ✓	✓	✗	✓	✓

3.5

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();   "Sub Class"
      }
   }
```

a. Sub Class

b. Super Class

[✓ c. Compiler Error] Cannot reach the private method.

d. Sub Class
   Super Class

2- What is the output of following Java program?

```
class Base {
   int i = 10;
}
class sub extends Base {
 ① int i = 20;
}
public class App2
{
      public static void main(String[]
 args){
          Base b = new sub();
          System.out.println(b.i);
      }          b = 20
}                 b.i
                   b
```

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

- 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(); // Child
                           child
      System.out.println(child.s);
                      // Child Child
   }
}
```

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) { =7-1
      this.i = i--; //6
   }
}

class B extends A{
   public B(int i) { // 6
      super(++i); //7
      System.out.println(i); // 6
   }
}

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

1) B(6) → super(7) → i-- = 6 في كلاس A

→ sy.o.p(i) =

في سب: $\boxed{7}$

في سوبر: ~~6~~

which one? ~~[illegible]~~

[✓ 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.

in java multipule inhert not Allowed.

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

a. True.

[✓ b. False.]

~~[illegible]~~

Super
↓
Sub
↓
Sub !

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.

Sub class يستطيع تغيير أو التأثير على ال private الموجودة في super فقط عن طريق public, protected الموجودة في سوبر كلاس وموروثة في ال Subclass ✓

Question#2: The following code will generate compiler error. Find out and correct the error in the below codes. *(3 pts.)*

3

1.

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

we need to call the Super constructor.

Error Correction:

Class B ~~have not~~ didn't Call the Superin the first Statment.

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

2.

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

It must Call Super(); in the first Statment.

Error Correction:

```
class B extends A { Public B(){
Super(); System.out.println
("Start of B"); System.out.println
("End of B"); }}
```

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:

((A) C) ~~c~~ = b;

4.

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

in B Class, Class A Cannot see it.

Error Correction:

the method MyMethod is only in "B" and not in "A".

((B) b).MyMethod();

there is No relationship between C & b.

~~the~~

we Can fix it by Casting the "C" ((A) C) = b;

or C c = new C(); ((A) c) = b;

```
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));
    }
}
```

we Cannot overwrite!

~~we Cannot~~ Overide a final method.

Error Correction:

Class A { public int Calculate (int a, Int b) {return a+b;}}

the error ~~is~~ because of the final method ~~that~~ were overide while it Cannot be overide.

Fixing it by ~~making~~ Changing the final method into a public one.

```
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();
   }
}
```

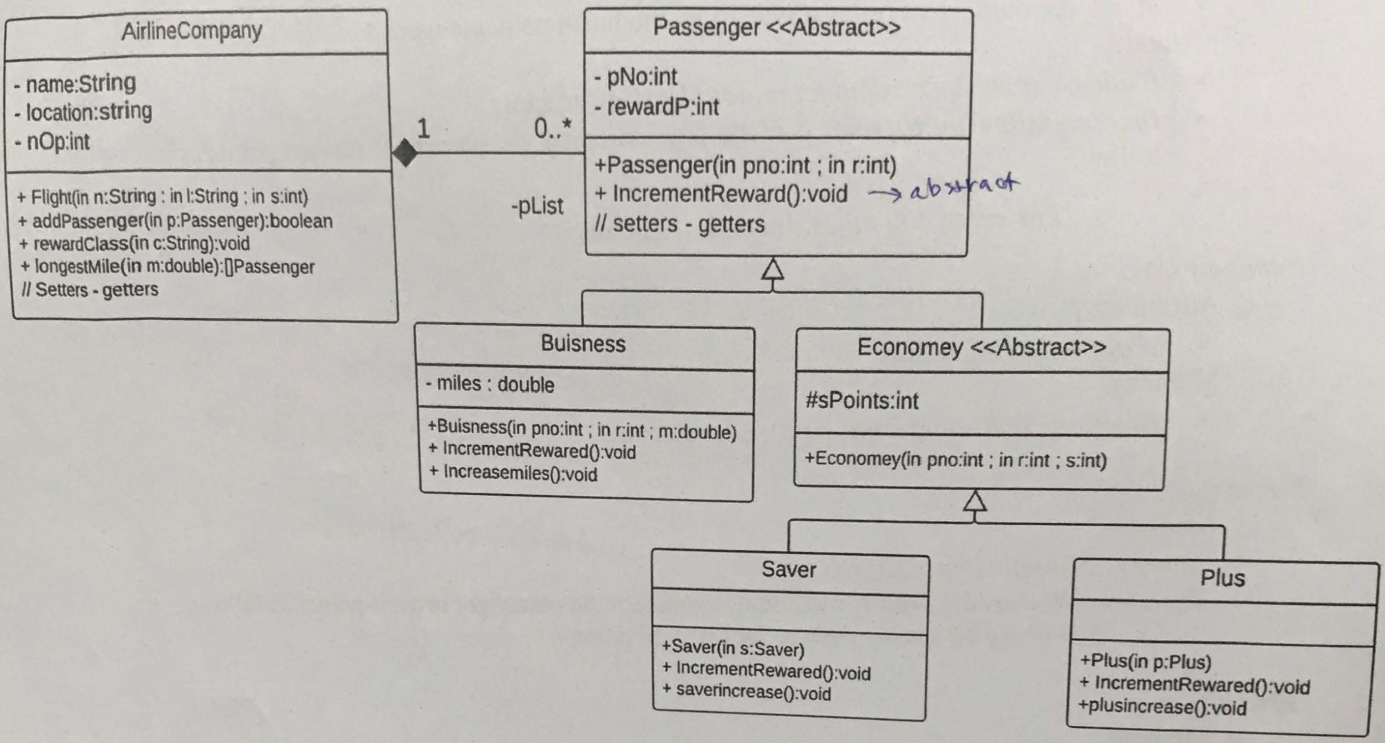
we cannot change the method type when we overide it! if i change from void to int i wanted to assign it to an int var.

Error Correction:

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

Fixing it by Changing the method type ---

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



<u>The descriptions of UML class:</u>

Passenger **class:**

- Attributes:
  - ***pNo***: the passport number of the passenger.
  - ***rewardP:*** passenger reward points.
- Methods:
  - ***Passenger(pNo:int; r: double)*:** Constructor.
  - ***IncrementRewared(): void:*** An <u>*abstract*</u> 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*. There is a maximum of only 50 business passengers on the company. 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*. t +
        - *Note*: the received class can be either Business or Economy.
    - ***longestMile(m:double):[] Passenger:*** this method returns an array *contains only* all **Business** passengers who have miles 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)` {

b) `public void rewardClass(String c)`

```
{
    ~~for(int i=0; i< nOp; i++)~~

        ~~if( pList[i] instanceof Business)~~
        ~~if (pList[i].getClass().getName().equals(c))~~
            ~~((Business) pList[i]).~~~~get~~ ~~IncrementReward();~~


    ~~for( int i=0; i<nOp; i++)~~
        ~~if( pList[i].getClass().getName().equals(c))~~

    ~~((pList[i].getClass().getName()) pList[i]).IncrementReward();~~

    for( int i=0; i<nOp; i++)
    if (pList [i].getClass().getName().equals(c))
    ((pList[i].getClass().getName()) pList[i]).IncrementReward();

}
```

```
c) public Passenger[] longestMile(double m)
{
        int Counter=0;   int index =0;
        for(int i=0; i<nOp; i++)
        if (pList[i] instanceof Business)
   if (((Business) pList[i]) .get Miles() > m)
             Counter++;

        Passenger[] longList = new Business[Counter];

  for(int i =0; i<nOp; i++)
  if( pList[i] instanceof Business)
  if(((Business)pList[i]) .getMiles()>m)
     ~~Retu~~
     LongList[index++]=new Business(((Business)pList[i]).getPNo(),
((Business) pList[i]).getReward p(),((Business)pList[i]).getMiles()
;

     if(Counter ==0) return null;
     return LongList;

  }
```

Cont.

getPNo
getRewarp
getMiles