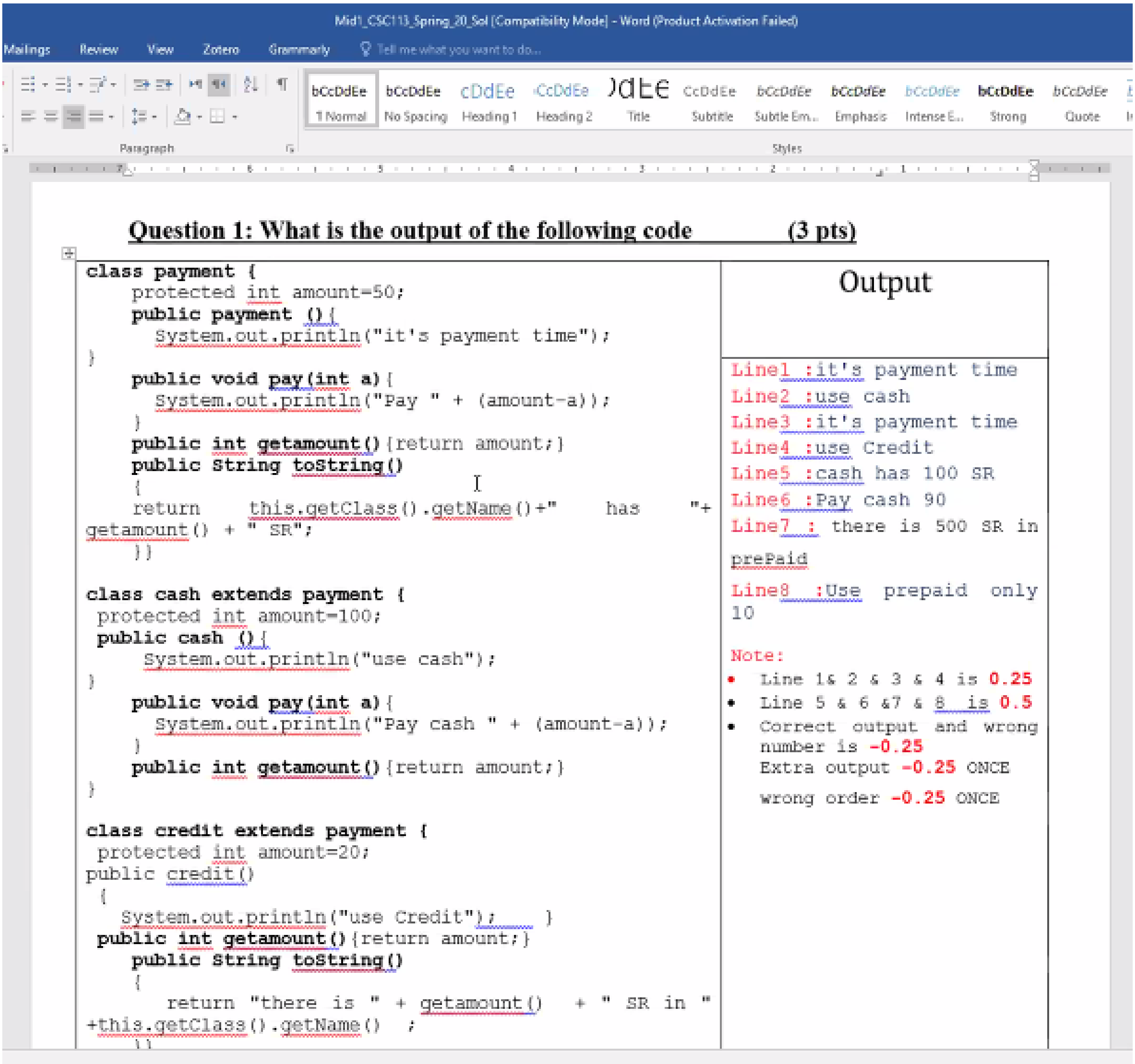


Question 1: What is the output of the following code (3 pts)

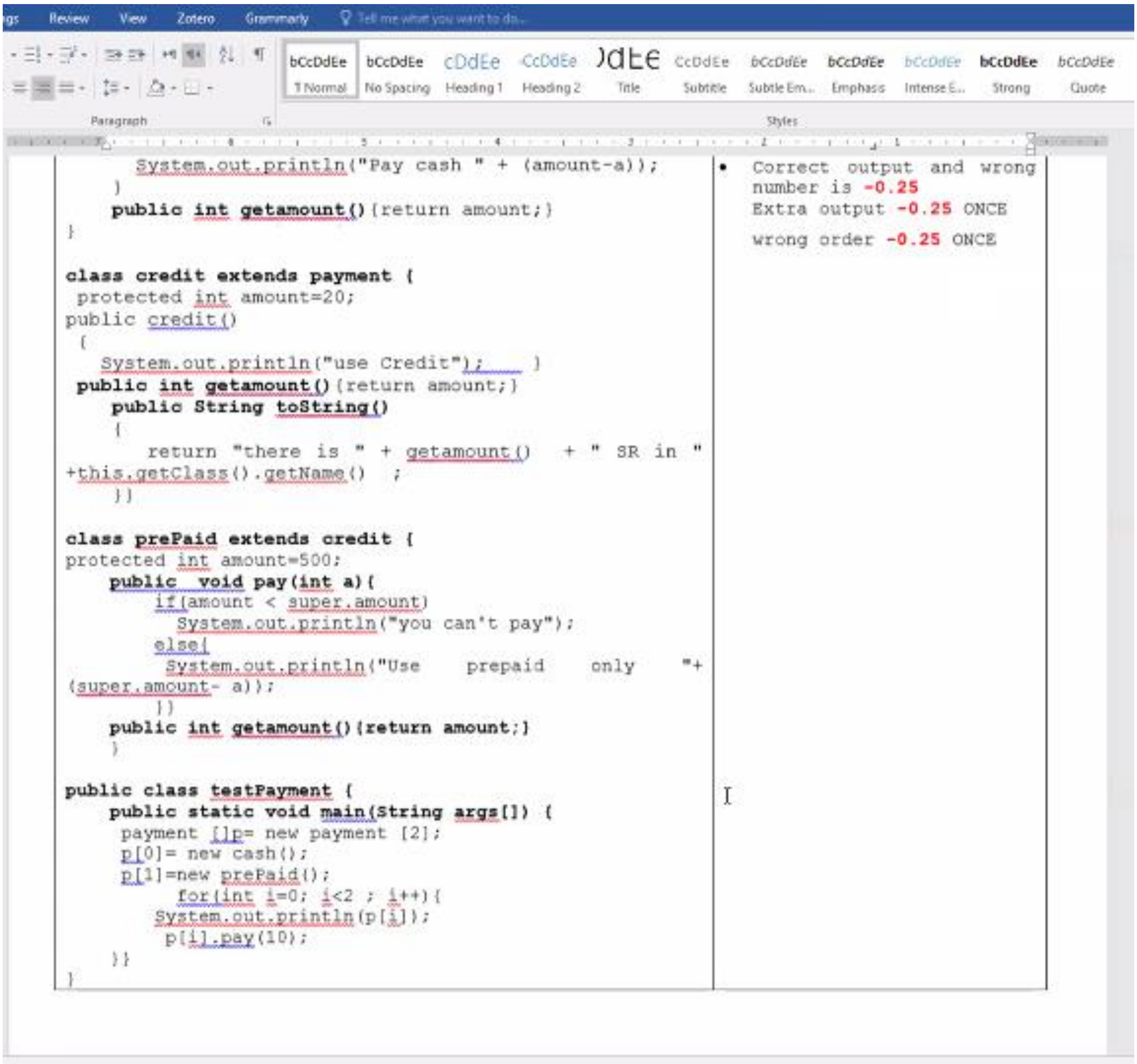
```
class payment {
    protected int amount=50;
    public payment (){
      System.out.println("it's payment time");
}
    public void pay(int a){
      System.out.println("Pay " + (amount-a));
    }
    public int getamount(){return amount;}
    public String toString()
    {
    return     this.getClass().getName()+"     has     "+
getamount() + " SR";
    }}

class cash extends payment {
 protected int amount=100;
 public cash (){
     System.out.println("use cash");
}
    public void pay(int a){
      System.out.println("Pay cash " + (amount-a));
    }
    public int getamount(){return amount;}
}

class credit extends payment {
 protected int amount=20;
public credit()
 {
   System.out.println("use Credit");     }
 public int getamount(){return amount;}
    public String toString()
    {
       return "there is " + getamount()   + " SR in "
+this.getClass().getName()  ;
    }}
```

Output

```
Line1 :it's payment time
Line2 :use cash
Line3 :it's payment time
Line4 :use Credit
Line5 :cash has 100 SR
Line6 :Pay cash 90
Line7 : there is 500 SR in
prePaid
Line8  :Use  prepaid  only
10
```

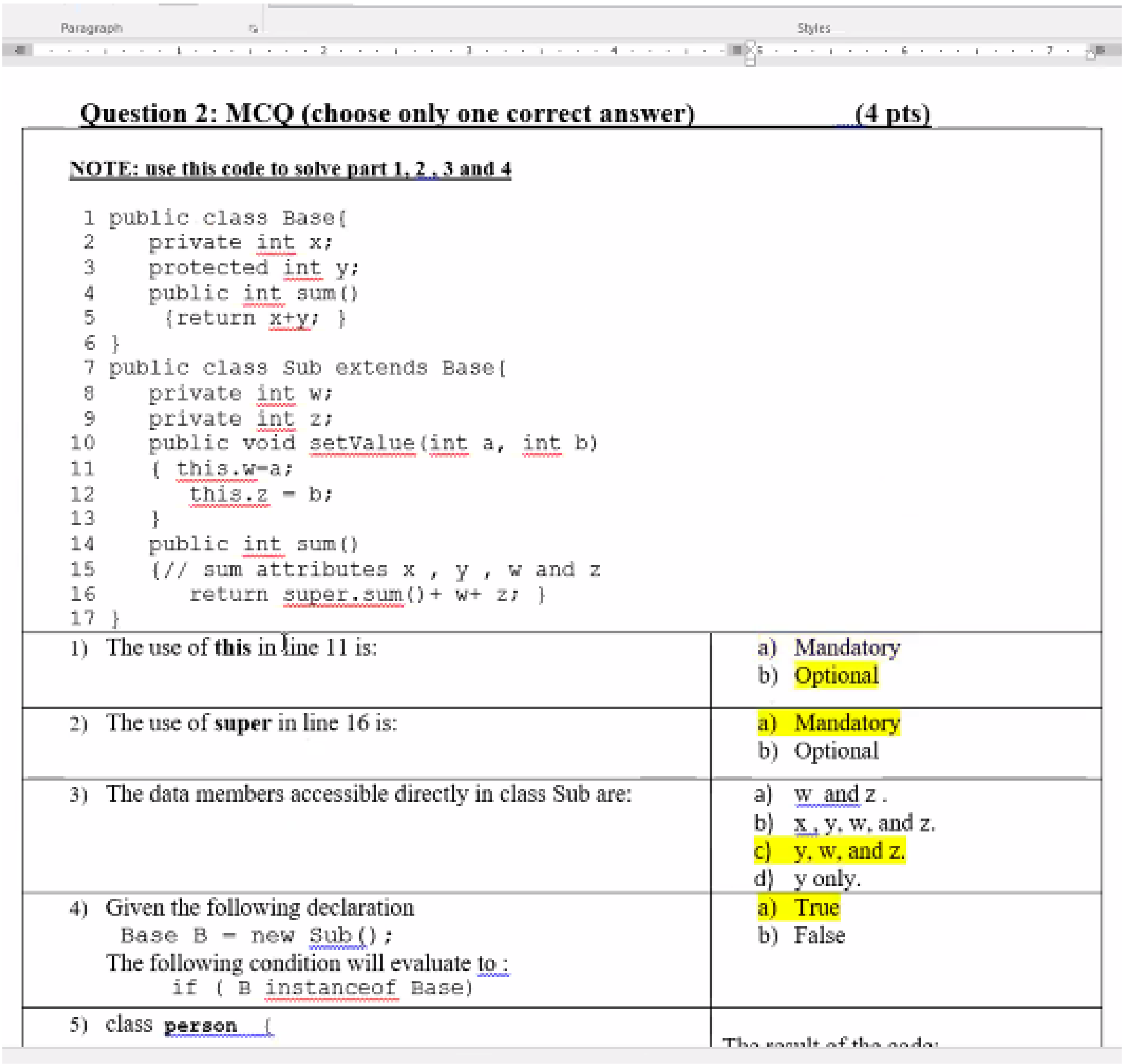
Note:

- Line 1& 2 & 3 & 4 is 0.25
- Line 5 & 6 &7 & 8 is 0.5
- Correct output and wrong number is -0.25
  Extra output -0.25 ONCE
  wrong order -0.25 ONCE



```
      System.out.println("Pay cash " + (amount-a));
    }
    public int getamount(){return amount;}
}

class credit extends payment {
 protected int amount=20;
public credit()
 {
   System.out.println("use Credit");    }
 public int getamount(){return amount;}
    public String toString()
    {
       return "there is " + getamount()   + " SR in "
+this.getClass().getName()   ;
    }}

class prePaid extends credit {
protected int amount=500;
    public  void pay(int a){
        if(amount < super.amount)
          System.out.println("you can't pay");
        else{
         System.out.println("Use    prepaid    only    "+
(super.amount- a));
        }}
    public int getamount(){return amount;}
    }

public class testPayment {
    public static void main(String args[]) {
     payment []p= new payment [2];
     p[0]= new cash();
     p[1]=new prePaid();
          for(int i=0; i<2 ; i++){
        System.out.println(p[i]);
         p[i].pay(10);
    }}
}
```

- Correct output and wrong number is **-0.25**
  Extra output **-0.25** ONCE
  wrong order **-0.25** ONCE



Question 2: MCQ (choose only one correct answer) (4 pts)

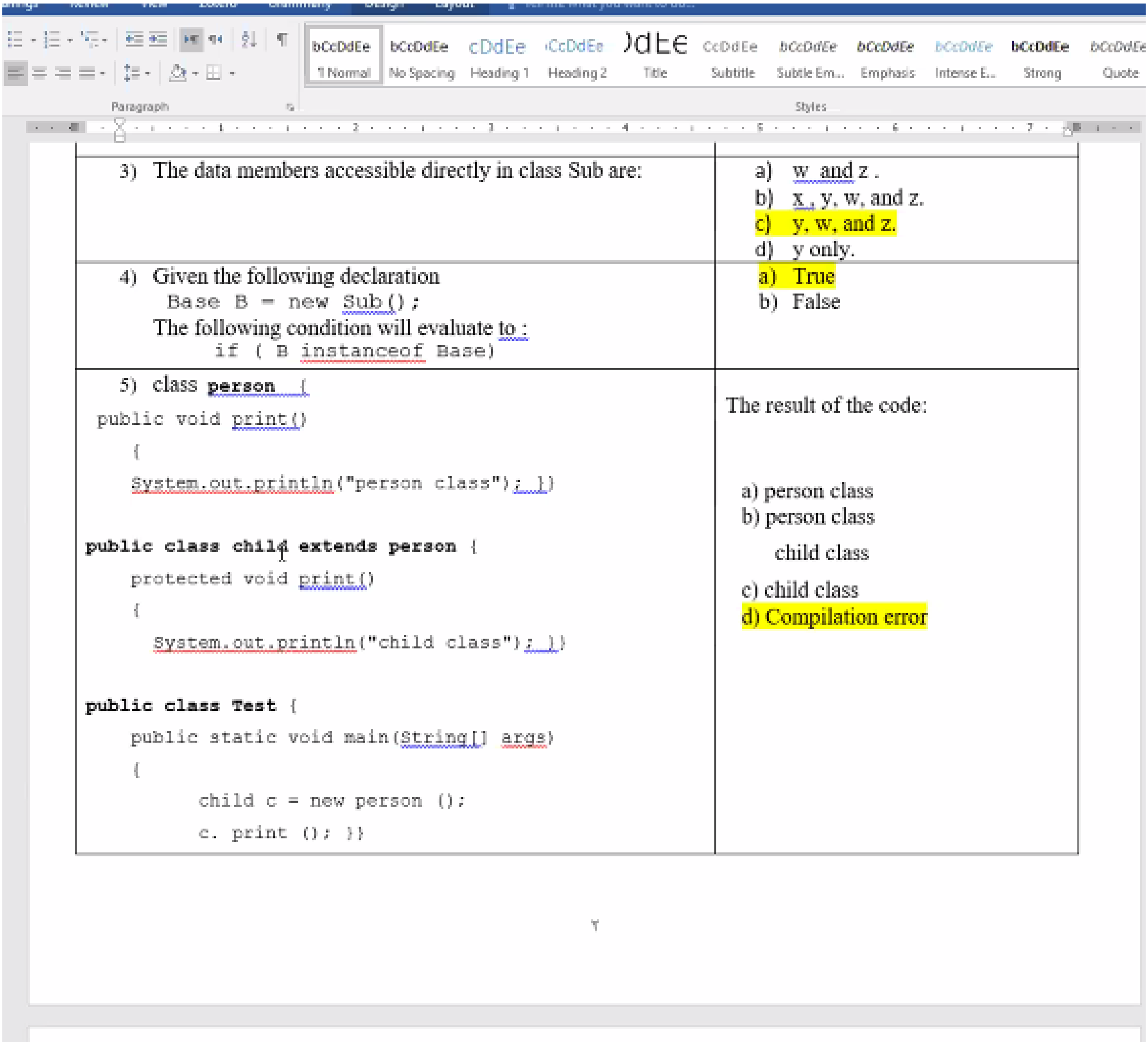
NOTE: use this code to solve part 1, 2 , 3 and 4

```
 1 public class Base{
 2    private int x;
 3    protected int y;
 4    public int sum()
 5     {return x+y; }
 6 }
 7 public class Sub extends Base{
 8    private int w;
 9    private int z;
10    public void setValue(int a, int b)
11    { this.w=a;
12       this.z = b;
13    }
14    public int sum()
15    {// sum attributes x , y , w and z
16       return super.sum()+ w+ z; }
17 }
```

1) The use of **this** in line 11 is:	a) Mandatory b) Optional
2) The use of **super** in line 16 is:	a) Mandatory b) Optional
3) The data members accessible directly in class Sub are:	a) w and z . b) x , y, w, and z. c) y, w, and z. d) y only.

4) Given the following declaration
`Base B = new Sub();`
The following condition will evaluate to :
`if ( B instanceof Base)`
	a) True
	b) False

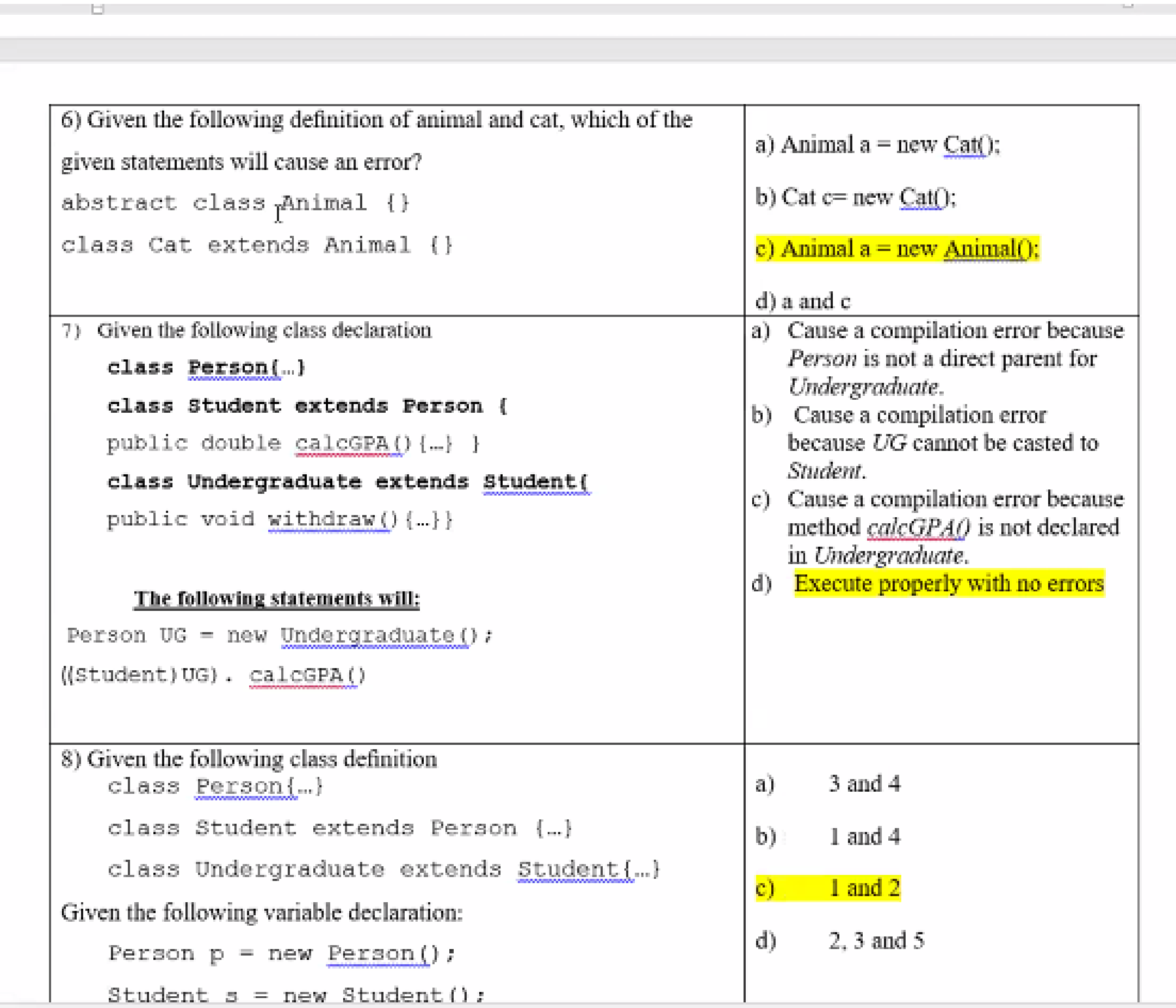
5) class **person** {	The result of the code:



3) The data members accessible directly in class Sub are:	a) w and z . b) x , y, w, and z. c) y, w, and z. d) y only.

4) Given the following declaration
`Base B = new Sub();`
The following condition will evaluate to :
`if ( B instanceof Base)`
	a) True
	b) False

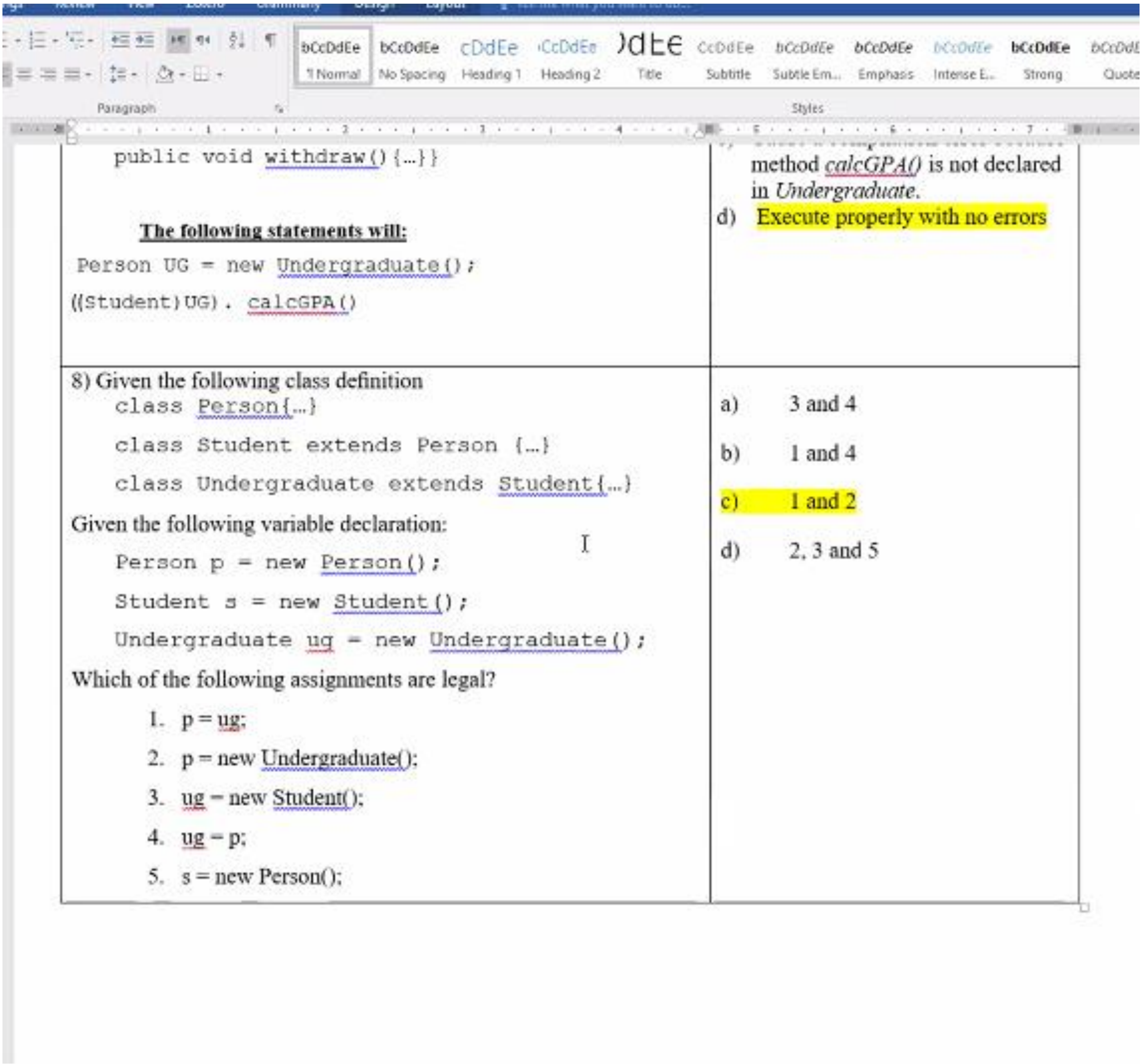
5) class **`person`** `{`
` public void print()`
`   {`
`   System.out.println("person class"); }}`
 	
`public class child extends person` `{`
`   protected void print()`
`   {`
`     System.out.println("child class"); }}`
 	
`public class Test` `{`
`   public static void main(String[] args)`
`   {`
`        child c = new person ();`
`        c. print (); }}`
	The result of the code:
	
	a) person class
	b) person class
	child class
	c) child class
	d) Compilation error



6) Given the following definition of animal and cat, which of the given statements will cause an error?
`abstract class Animal {}`
`class Cat extends Animal {}`
	a) Animal a = new Cat();
	b) Cat c= new Cat();
	c) Animal a = new Animal();
	d) a and c

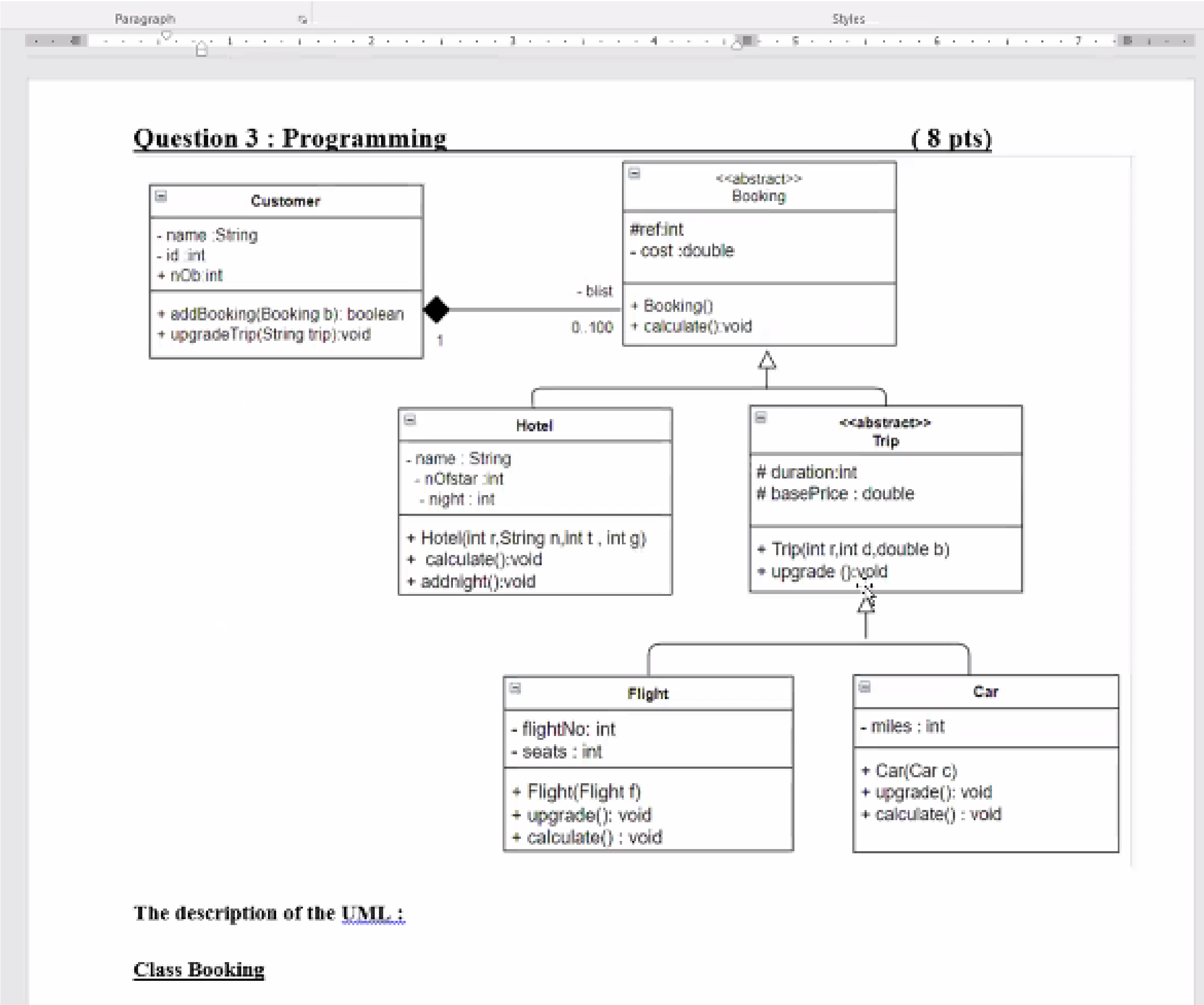
7) Given the following class declaration
`class Person{…}`
`class Student extends Person {`
`public double calcGPA(){…} }`
`class Undergraduate extends Student{`
`public void withdraw(){…}}`
<u>The following statements will:</u>
`Person UG = new Undergraduate();`
`((Student)UG). calcGPA()`
	a) Cause a compilation error because *Person* is not a direct parent for *Undergraduate*.
	b) Cause a compilation error because *UG* cannot be casted to *Student*.
	c) Cause a compilation error because method *calcGPA()* is not declared in *Undergraduate*.
	d) Execute properly with no errors

8) Given the following class definition
`class Person{…}`
`class Student extends Person {…}`
`class Undergraduate extends Student{…}`
Given the following variable declaration:
`Person p = new Person();`
`Student s = new Student();`
	a) 3 and 4
	b) 1 and 4
	c) 1 and 2
	d) 2, 3 and 5



public void withdraw(){…}}
The following statements will:
Person UG = new Undergraduate();
((Student)UG). calcGPA()
	method *calcGPA()* is not declared in *Undergraduate.*
	d) Execute properly with no errors

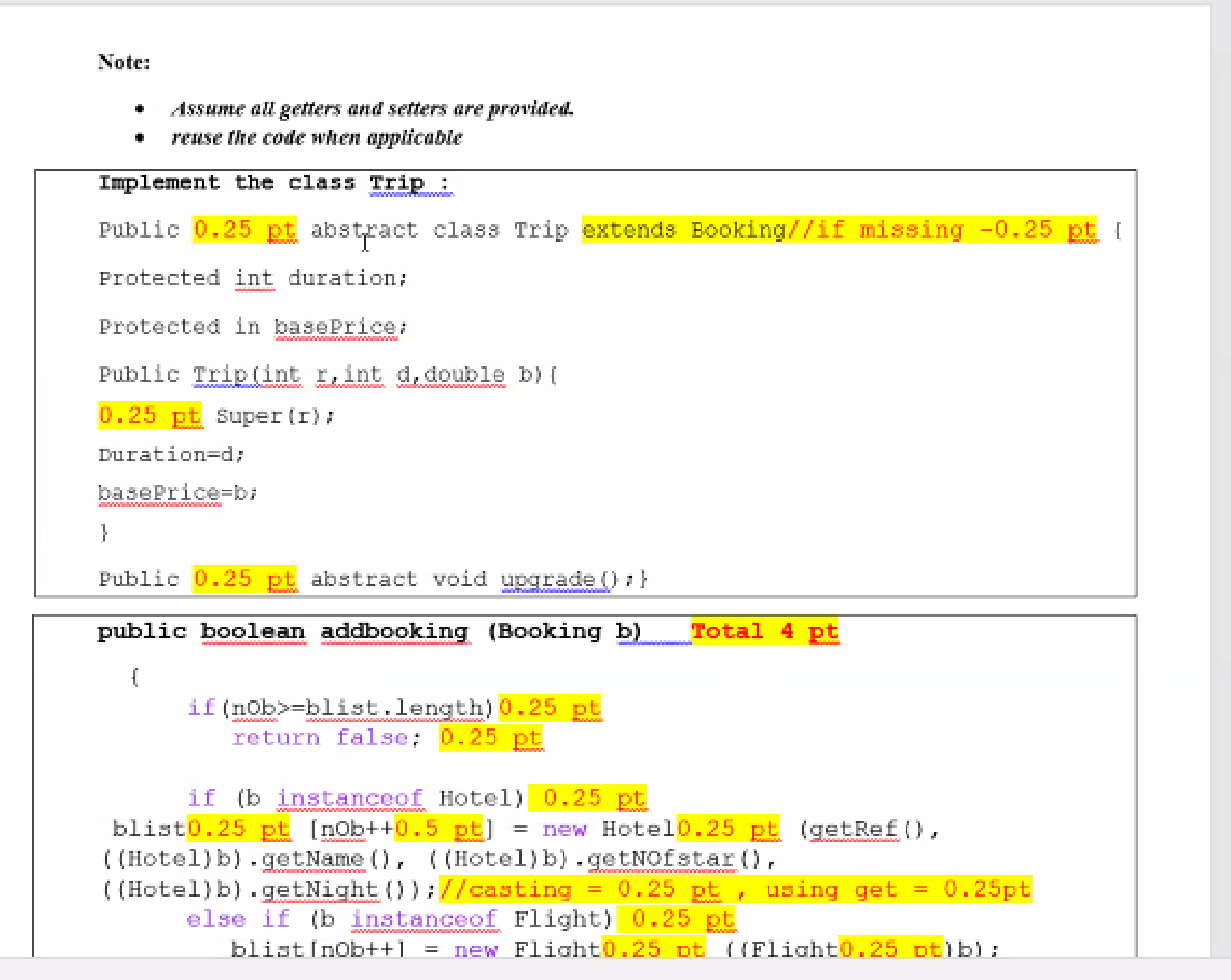
8) Given the following class definition class Person{…} class Student extends Person {…} class Undergraduate extends Student{…} Given the following variable declaration: Person p = new Person(); Student s = new Student(); Undergraduate ug = new Undergraduate(); Which of the following assignments are legal? 1. p = ug; 2. p = new Undergraduate(); 3. ug = new Student(); 4. ug = p; 5. s = new Person();	a) 3 and 4 b) 1 and 4 c) 1 and 2 d) 2, 3 and 5



Question 3 : Programming (8 pts)

The description of the UML :

Class Booking

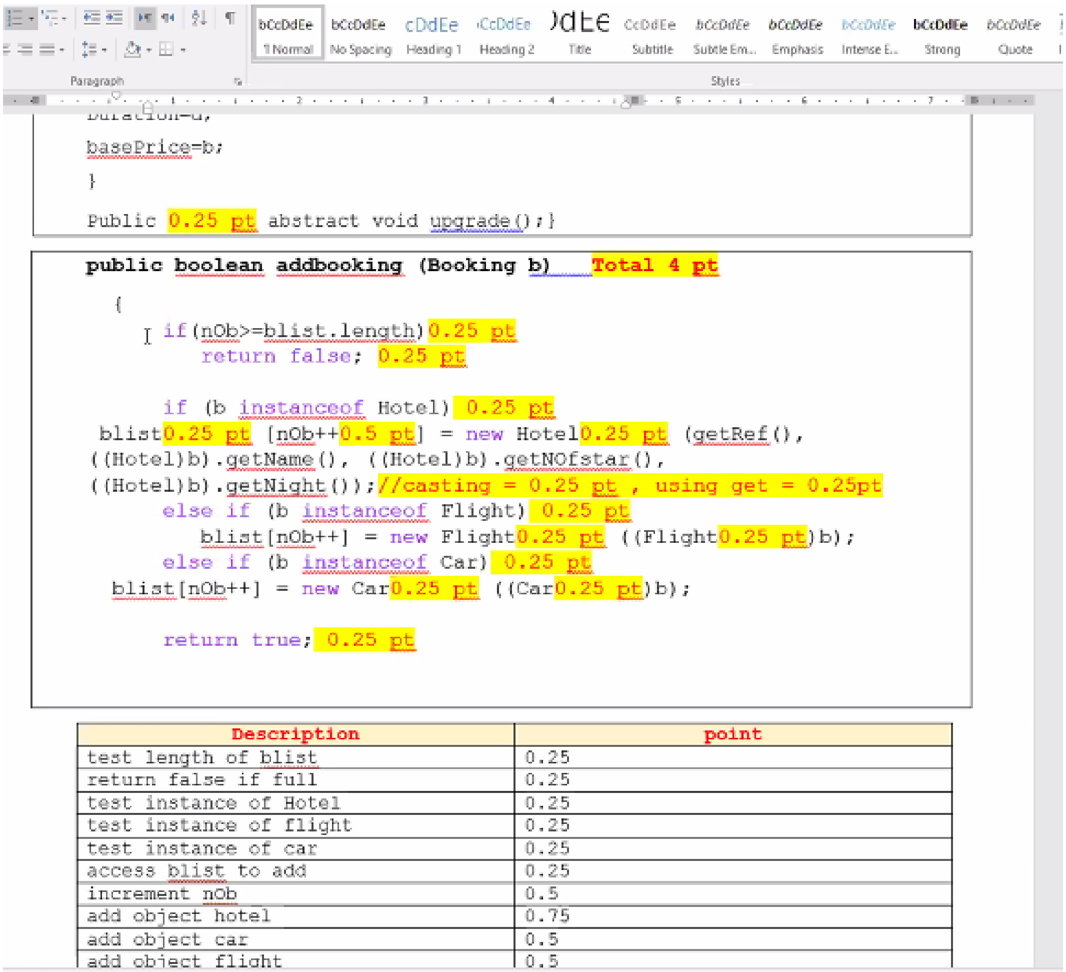


Note:

- *Assume all getters and setters are provided.*
- *reuse the code when applicable*

```
Implement the class Trip :

Public 0.25 pt abstract class Trip extends Booking//if missing -0.25 pt {

Protected int duration;

Protected in basePrice;

Public Trip(int r,int d,double b){

0.25 pt Super(r);

Duration=d;

basePrice=b;

}

Public 0.25 pt abstract void upgrade();}
```

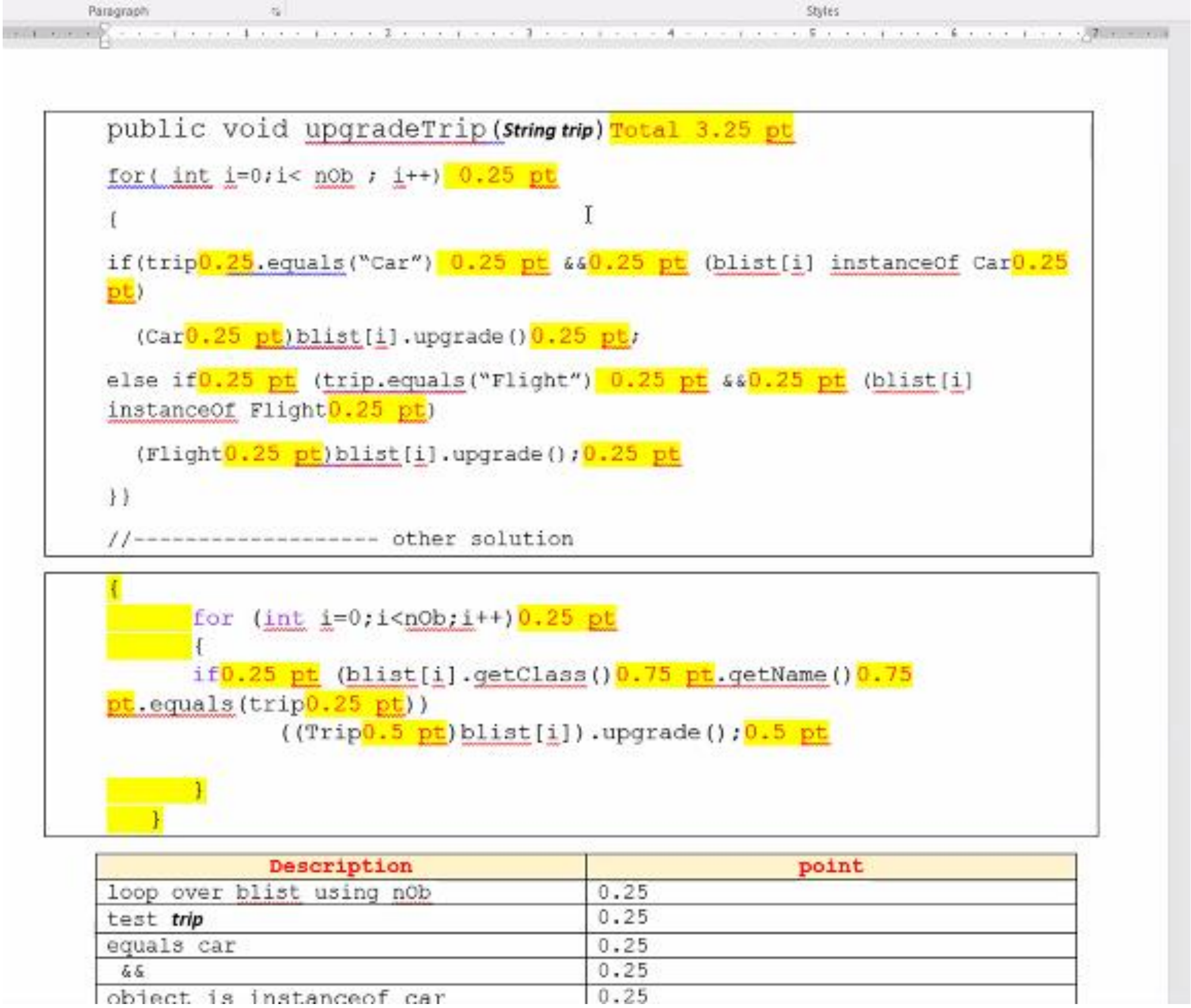
```
public boolean addbooking (Booking b)    Total 4 pt

  {
      if(nOb>=blist.length)0.25 pt
         return false; 0.25 pt

      if (b instanceof Hotel) 0.25 pt
 blist0.25 pt [nOb++0.5 pt] = new Hotel0.25 pt (getRef(),
((Hotel)b).getName(), ((Hotel)b).getNOfstar(),
((Hotel)b).getNight());//casting = 0.25 pt , using get = 0.25pt
      else if (b instanceof Flight) 0.25 pt
         blist[nOb++] = new Flight0.25 pt ((Flight0.25 pt)b);
```



```
Duration=d;
basePrice=b;
}
Public 0.25 pt abstract void upgrade();}
```

```
public boolean addbooking (Booking b)    Total 4 pt
  {
    if(nOb>=blist.length)0.25 pt
       return false; 0.25 pt

    if (b instanceof Hotel) 0.25 pt
 blist0.25 pt [nOb++0.5 pt] = new Hotel0.25 pt (getRef(),
((Hotel)b).getName(), ((Hotel)b).getNOfstar(),
((Hotel)b).getNight());//casting = 0.25 pt , using get = 0.25pt
    else if (b instanceof Flight) 0.25 pt
       blist[nOb++] = new Flight0.25 pt ((Flight0.25 pt)b);
    else if (b instanceof Car) 0.25 pt
  blist[nOb++] = new Car0.25 pt ((Car0.25 pt)b);

    return true; 0.25 pt
```

Description	point
test length of blist	0.25
return false if full	0.25
test instance of Hotel	0.25
test instance of flight	0.25
test instance of car	0.25
access blist to add	0.25
increment nOb	0.5
add object hotel	0.75
add object car	0.5
add object flight	0.5



```
public void upgradeTrip(String trip) Total 3.25 pt
for( int i=0;i< nOb ; i++) 0.25 pt
{
if(trip0.25.equals("Car") 0.25 pt &&0.25 pt (blist[i] instanceOf Car0.25
pt)
  (Car0.25 pt)blist[i].upgrade()0.25 pt;
else if0.25 pt (trip.equals("Flight") 0.25 pt &&0.25 pt (blist[i]
instanceOf Flight0.25 pt)
  (Flight0.25 pt)blist[i].upgrade();0.25 pt
}}
//------------------- other solution
```

```
{
        for (int i=0;i<nOb;i++)0.25 pt
        {
        if0.25 pt (blist[i].getClass()0.75 pt.getName()0.75
pt.equals(trip0.25 pt))
                ((Trip0.5 pt)blist[i]).upgrade();0.5 pt

        }
    }
```

Description	point
loop over blist using nOb	0.25
test ***trip***	0.25
equals car	0.25
&&	0.25
object is instanceof car	0.25