United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Mid Exam :: Fall 2021

Course Code: CSE 1115 **Course Title: Object Oriented Programming**

Total Marks: **30** Time: **1hr 45 mins**

READ THIS CAREFULLY: Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules

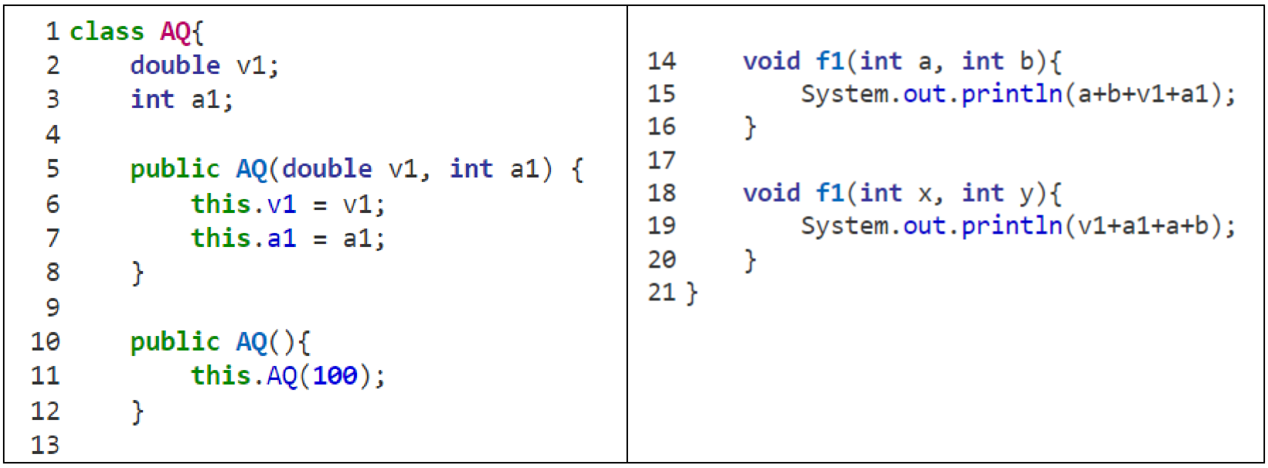
Answer all the SIX questions

Question 1 [5]

Write a java program to print the area of a rectangle by creating a class named ‘*Area*’ taking the values of its ‘*length*’ and ‘*width*’ as private instance variables of integer type. The constructor initializes the instance variables with ‘this’ reference keyword. Area has a method named ‘*returnArea*’ which returns the area of the rectangle. Length and width of rectangle are entered through the object creation that’s done in the ‘*main*’ method. There’s a block in this class that prints the following message ‘The area A of a rectangle is given by the formula, A=lw, where l is the length and w is the width.’.

Question 2 [2 + 3]

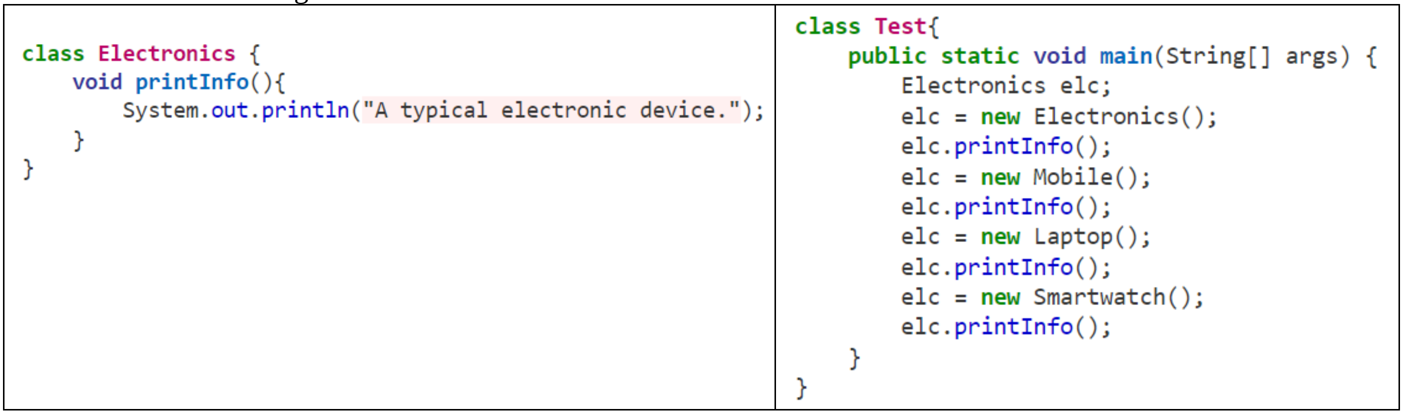
A. Consider the following AQ class:



```
 1 class AQ{
 2     double v1;
 3     int a1;
 4
 5     public AQ(double v1, int a1) {
 6         this.v1 = v1;
 7         this.a1 = a1;
 8     }
 9
10     public AQ(){
11         this.AQ(100);
12     }
13
14     void f1(int a, int b){
15         System.out.println(a+b+v1+a1);
16     }
17
18     void f1(int x, int y){
19         System.out.println(v1+a1+a+b);
20     }
21 }
```

The code contains some errors. Fix the errors and write the correct code. You cannot remove any lines from the code. Note: You do not need to rewrite the whole code. You can use line numbers to fix errors in that line only.

B. Consider the following code:



```
class Electronics {
    void printInfo(){
        System.out.println("A typical electronic device.");
    }
}
```

```
class Test{
    public static void main(String[] args) {
        Electronics elc;
        elc = new Electronics();
        elc.printInfo();
        elc = new Mobile();
        elc.printInfo();
        elc = new Laptop();
        elc.printInfo();
        elc = new Smartwatch();
        elc.printInfo();
    }
}
```

Write the necessary **missing codes** so that the following output is produced when the program (Test.main) runs:

```
A typical electronic device.
A mobile is a portable telephone.
A laptop is a small, portable personal computer.
A smartwatch is a wearable computer in the form of a watch.
```

Question 3 [3 + 2]

Consider the following code:

```
 1 class Exam {
 2     int id;
 3     double mark;
 4 }
 5
 6 class MTest{
 7     void testFnc1(Exam m1, Exam m2){
 8         Exam temp = m1;
 9         m1 = m2;
10         m2 = temp;
11     }
12
13     void testFnc2(Exam m1, Exam m2){
14         int tempId = m1.id;
15         double tempMark = m2.mark;
16         m1.id = m2.id;   m1.mark = m2.mark;
17         m2.id = tempId;  m2.mark = tempMark;
18     }
19
```

```
20     public static void main(String[] args) {
21         MTest testObj = new MTest();
22         Exam e1 = new Exam();
23         Exam e2 = new Exam();
24         e1.id = 10;     e1.mark = 70;
25         e2.id = 20;     e2.mark = 85;
26         System.out.println(e1.id + " " + e1.mark);
27         testObj.testFnc1(e1, e2);
28         System.out.println(e1.id + " " + e1.mark);
29         testObj.testFnc2(e1, e2);
30         System.out.println(e1.id + " " + e1.mark);
31
32         e1 = new Exam();
33         e2 = new Exam();
34     }
35 }
```

A. Here, in the main method, we created 2 objects of the class Exam and passed those objects to the testFnc1 and testFnc2 methods of the MTest class. Write the output for the println calls in lines 26, 28, and 30. Note that, only e1's id and mark values are printed.

B. Explain what will happen after lines 32 and 33 are executed in terms of garbage collection. Keep your answer short and precise.

Question 4 [5]

Construct a class named **Member**. The Member class has five private instance variables namely, name (String), age (int), number (String), salary (double), experience (double). The Member class must have the followings:

i. A constructor that initializes all five instance variables.

ii. Write appropriate setter-getter methods for each instance variables.

Construct a new class named **Employee** that inherits Member class. The Employee class has its own attributes named designation (String). The Employee class must have the followings:

iii. A constructor that gets all the Member class variables using super constructor and initializes the designation variable.

iv. A changedesignation() method that changes the designation of an employee if the experience is greater than five. If the change is not possible then print a simple message stating that designation change is not possible.

v. A getdesignation() method that returns the designation of the employee.

Construct a **Mainclass**. Inside the Mainclads, create an object of Employee with proper constructor call. Provide a call to changedesignation() method to check if the designation is changed or not.

Question 5 [2 + 3]

A. Find the output of the following java code:

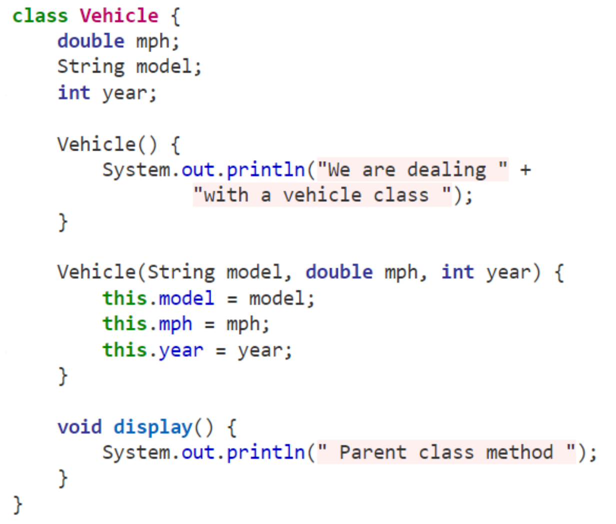
```
class Parent {
    int i = 10;

    public Parent(int j) {
        System.out.println(i);
        j = j * 2;
        this.i = j * 10;
    }
}

class Child extends Parent {
    public Child(int j) {
        super(j);
        System.out.println(i);
        this.i = j * 20;
    }
}
```

```
public class MainClass {
    public static void main(String[] args) {
        Child n = new Child(4);
        System.out.println(n.i);
    }
}
```

B. Study the following code below and write down necessary lines in the highlighted area:

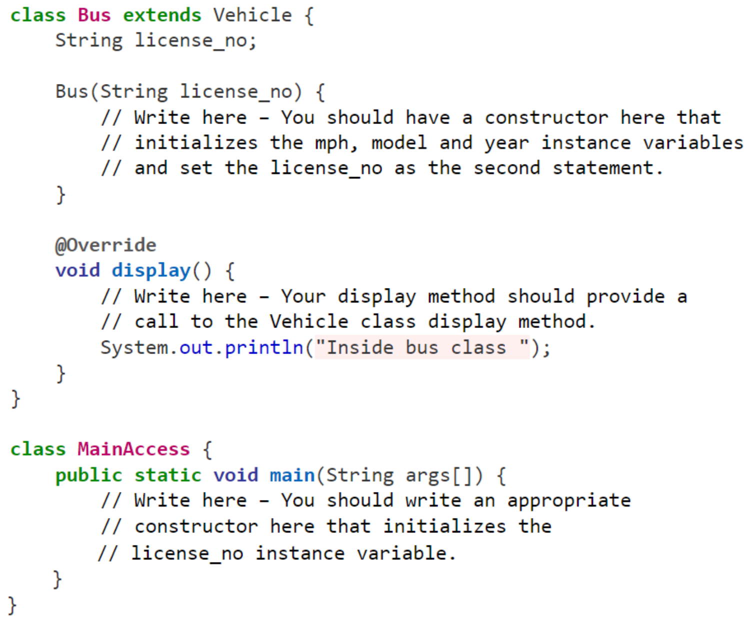


```
class Vehicle {
    double mph;
    String model;
    int year;

    Vehicle() {
        System.out.println("We are dealing " +
                "with a vehicle class ");
    }

    Vehicle(String model, double mph, int year) {
        this.model = model;
        this.mph = mph;
        this.year = year;
    }

    void display() {
        System.out.println(" Parent class method ");
    }
}
```



```
class Bus extends Vehicle {
    String license_no;

    Bus(String license_no) {
        // Write here - You should have a constructor here that
        // initializes the mph, model and year instance variables
        // and set the license_no as the second statement.
    }

    @Override
    void display() {
        // Write here - Your display method should provide a
        // call to the Vehicle class display method.
        System.out.println("Inside bus class ");
    }
}

class MainAccess {
    public static void main(String args[]) {
        // Write here - You should write an appropriate
        // constructor here that initializes the
        // license_no instance variable.
    }
}
```

Question 6 [2+2+1]

Consider the abstract class "Item" that has a constructor that takes arguments:
(name: *String*, unit_price: *double*) to set item name & unit price.
Now complete the following three tasks (***No need to rewrite the codes in the question. Just write the answer parts only***):

a. Write a class named "WeightedItem" extending the "Item" class.

i. Add **one additional field** named "weight" that holds the weight of this item

ii. Add a **constructor** that takes three arguments: (name: *String*, weight: *double*, unit_price: *double*)

iii. Now **override** the method "calculatePrice" that should return total price for this item object as:

weight * unit_price

b. Write another class named "CountableItem" extending the "Item" class.

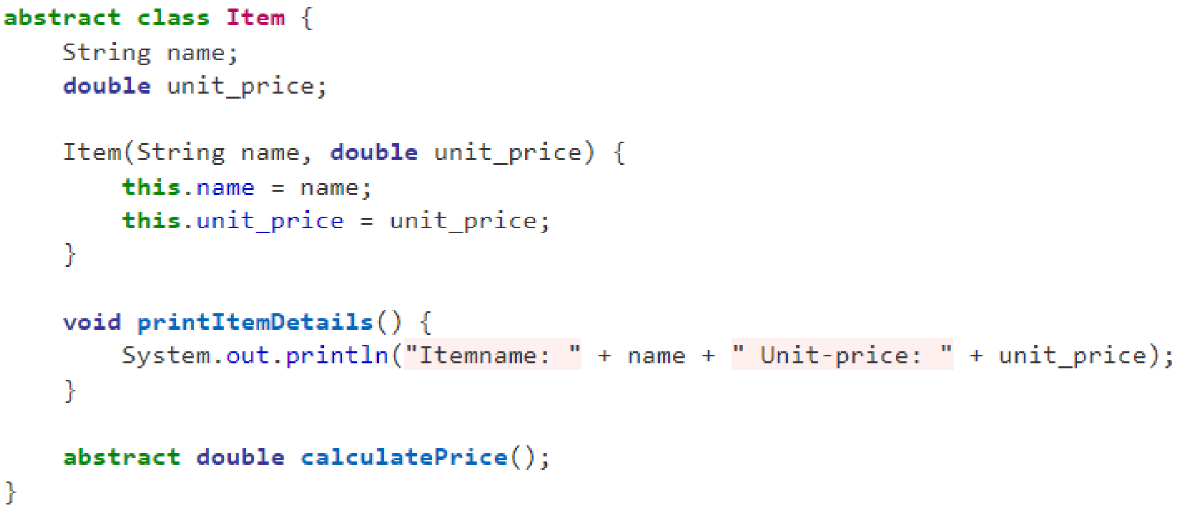
i. Add **one additional field** named "quantity" that holds the quantity as count of this item

ii. Add a **constructor** that takes three arguments: (name: *String*, quantity: *int*, unit_price: *double*)

iii. Now **override** the method "calculatePrice" that should return total price for this item object as:

quantity * unit_price

c. Now complete the "calculateTotalPrice" method that takes an array of Items and returns the calculated total price for all the items in that array using **for each loop.**

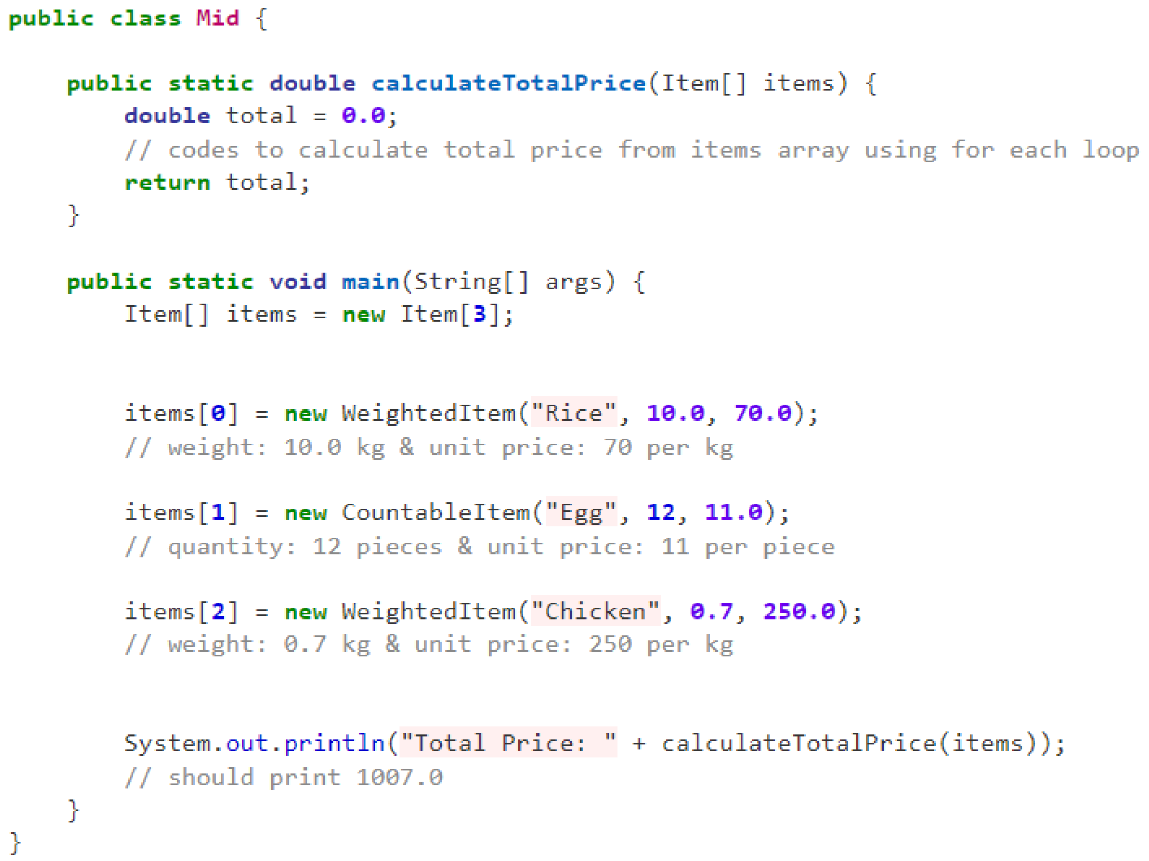


```
abstract class Item {
    String name;
    double unit_price;

    Item(String name, double unit_price) {
        this.name = name;
        this.unit_price = unit_price;
    }

    void printItemDetails() {
        System.out.println("Itemname: " + name + " Unit-price: " + unit_price);
    }

    abstract double calculatePrice();
}
```



```java
public class Mid {

    public static double calculateTotalPrice(Item[] items) {
        double total = 0.0;
        // codes to calculate total price from items array using for each loop
        return total;
    }

    public static void main(String[] args) {
        Item[] items = new Item[3];


        items[0] = new WeightedItem("Rice", 10.0, 70.0);
        // weight: 10.0 kg & unit price: 70 per kg

        items[1] = new CountableItem("Egg", 12, 11.0);
        // quantity: 12 pieces & unit price: 11 per piece

        items[2] = new WeightedItem("Chicken", 0.7, 250.0);
        // weight: 0.7 kg & unit price: 250 per kg


        System.out.println("Total Price: " + calculateTotalPrice(items));
        // should print 1007.0
    }
}
```