United International University (UIU)
Dept. of Computer Science & Engineering (CSE)
Mid Exam :: Summer 2021
Course Code: CSE 1115   Course Title: Object Oriented Programming
Total Marks: **20**   Time: **1hr**

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

Question 1 (4) [CO1]

Consider the following code and answer the questions:

```java
package com.example.test;

public class Address {
    public String city;
    public int postCode;
}
```

```java
package com.example.test1;

public class Student extends Person{
    public  void print(){
        // prints the value of name, age, address
    }
}
```

```java
package com.example.test;

public class Person{
    private String name;
    private int age;
    private Address address;

    Person(){
        name = "Tamim Iqbal";
        age = 35;
        address = new Address();
        address.postCode = 1200;
        address.city = "Dhaka";
    }
}
```

a. Suggest two modifications in the Person class so that the class variables can be accessible from Student class. You are not allowed to use the **public** keyword for the variables.   **[2]**
b. Add the necessary code lines in Student class so that it can access variables in Address and Person classes.   **[2]**

Question 2 (4) [CO1]

Consider the following code:

```java
class Book{
    public int chapters;
    public int pages;
    public Book(int chapters, int pages){
        this.chapters = chapters;
        this.pages = pages;
    }
    public void printSummary(){
        System.out.println("Chapters: " + chapters);
        System.out.println("Pages: " + pages);
    }
}

class EBook extends Book{
    public String format;
}
```

```java
public class InheritanceTest{
    public static void main(String[] args) {
        EBook java = new EBook();
        int chapters = 8, pages = 500;
        EBook c = new EBook(chapters, pages, "pdf");
        Book b = java;
        b.printSummary();
        b = c;
        b.printSummary();
    }
}
```

Class EBook that extends or inherits the class Book. EBook has different types of formats: pdf/epub/mobi, etc. In the code, an incomplete version of the EBook class is given.

The main method inside the InheritanceTest class is used to test the classes (Book/EBook).

Now, Complete the EBook Class by creating **one or more constructors** and **overriding methods if needed**. You need to rewrite the EBook class only. You cannot change any other classes. Consider all three classes within the same package. Also **write** the **output** of the main method.

Question 3 (4) [CO1]

Suppose your brother owns an online shop and recently he decided to give discounts on some of the items in the shop. Different items can have different discounts, i.e., 5% discount, 10% discount etc. Now, your brother has sold some **discounted** items (let's say 5 items). Now he wants your help in finding the **total sales value** of the items sold.

To help you with the task, the Item class and the DiscountedItem class (partial) is written for you.

```java
class Item {
    String name;
    double price;

    public double getPrice() {
        return price;
    }
}

class Shop{
    public static void main(String[] args) {
        Item[] items = new Item[5];

        // Code 2: Initialize itmes array and
        // calculate the total price

    }
}
```

```java
class DiscountedItem extends Item{
    double discount;

    DiscountedItem(String name, double price,
                        double discount){
        this.name = name;
        this.price = price;
        this.discount = discount;
    }

    public double getPrice() {
        // Code 1: Calculate and return
        // the discounted price
    }
}
```

You will need to:
a. Complete the DiscountedItem class by writing the code inside the **getPrice** method. The getPrice method in the DiscountedItem class should calculate and return the discounted price using **price** and **discount** instance variables.
b. Initialize the array **items** in the **main** method. The array should contain DiscountedItem objects. After that, **use the array to calculate the total sales value**.

[**Note**: You can only write the necessary codes, with the heading **Code 1** & **Code 2**.]

Question 4 (4) [CO1]

Write a Java program to accomplish each of the following tasks (Each task has 0.5 marks):
a) Create a package named *"movies"*.
b) In *"movies"* package, create a Class named *"Titanic"* which has the *main* method.
c) In *"Titanic"* class, declare two instance variables named *"directorName"* and *"featuredSong"* of String type.
d) Write the constructor for the *"Titanic"* class and initialized the instance variables using this reference keyword.
e) Write a block in *"Titanic"* class to print *"Titanic sank in the North Atlantic Ocean on 15 April 1912 after striking an iceberg"*.
f) Write a method named *"showDetails"* that prints the values of instances variables.
g) Create an object of *"Titanic"* class in the *main* method of *"Titanic"* class and pass the following Strings by the constructor: "James Cameron" and "My Heart Will Go On".
h) Call the *"showDetails"* method from the *main* method of *"Titanic"* class.

Question 5 (4) [CO1]

Consider you working as a backend developer in a team for developing a game named *Dhaka Road Simulator*. Now your front-end developer of your team wants the following output of the right side whenever she will execute the code on the left side. Your team lead is very strict about object oriented design.

Solution:

```java
// Vehicle as an abstract class (common abstraction)
abstract class Vehicle {
    // move() differs for every vehicle -> abstract
    public abstract void move();

    // permission() is common to all vehicles -> concrete (no output needed)
    public void permission() {
        // common implementation
    }
}

// MotorVehicle adds engine-related behaviour
abstract class MotorVehicle extends Vehicle {
    public MotorVehicle() {
        System.out.println("Burning engine fuel");
    }

    // common implementation for all motor vehicles
    public void refill_tank() {
        System.out.println("Filled with fuel");
    }
}

// Rickshaw is a non-motor Vehicle
class Rickshaw extends Vehicle {
    public Rickshaw() {
        System.out.println("Paddling the chain");
    }

    @Override
    public void move() {
        System.out.println("Rickshaw can move in narrow streets");
    }
}

class CNG extends MotorVehicle {
    @Override
    public void move() {
        System.out.println("CNG can move in city streets");
    }
}

class Car extends MotorVehicle {
    @Override
    public void move() {
        System.out.println("Car can move in highway");
    }
}

class Bus extends MotorVehicle {
    @Override
    public void move() {
        System.out.println("Bus can move in expressway");
    }
}
```