



# COMSATS University Islamabad

Department of Computer Science

Midterm Examination Fall 2021

Object Oriented Programming

Class: BSCS A&B/BSSE A&B

Reg#:

24

Marks: 50

Date: 16-11-2021

Note: Attempt all the questions on the answer sheet. Lead pencils and markers are not allowed to use. In case of overwriting no marks will be awarded.

1. Consider the following class Person. It is a mutable class. Make necessary changes in class Person to make it immutable. On answer sheet write its immutable version. [CLO-1] [Marks: 10]

```
import java.util.Scanner;
public class Person {
    private String name;
    public int age;
    private Address adr;
    public Person() {
        name=null;
        age=0;
        adr=null;
    }
    public Person(String name, int age, Address adr) {
        this.name=name;
        this.age=age;
        this.adr=adr;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public Address getAddress() {
        return adr;
    }
    public void readData() {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter your name:");
        name=input.nextLine();
        System.out.println("Enter your age:");
        age=input.nextInt();
        System.out.println("Enter your house no:");
        adr.setHouseNo(input.nextInt());
        System.out.println("Enter your street:");
        adr.setStreet(input.nextLine());
        System.out.println("Enter your city:");
        adr.setCity(input.nextLine());
    }
}
```

```
public class Address {
    private int houseNo;
    private String street;
    private String city;

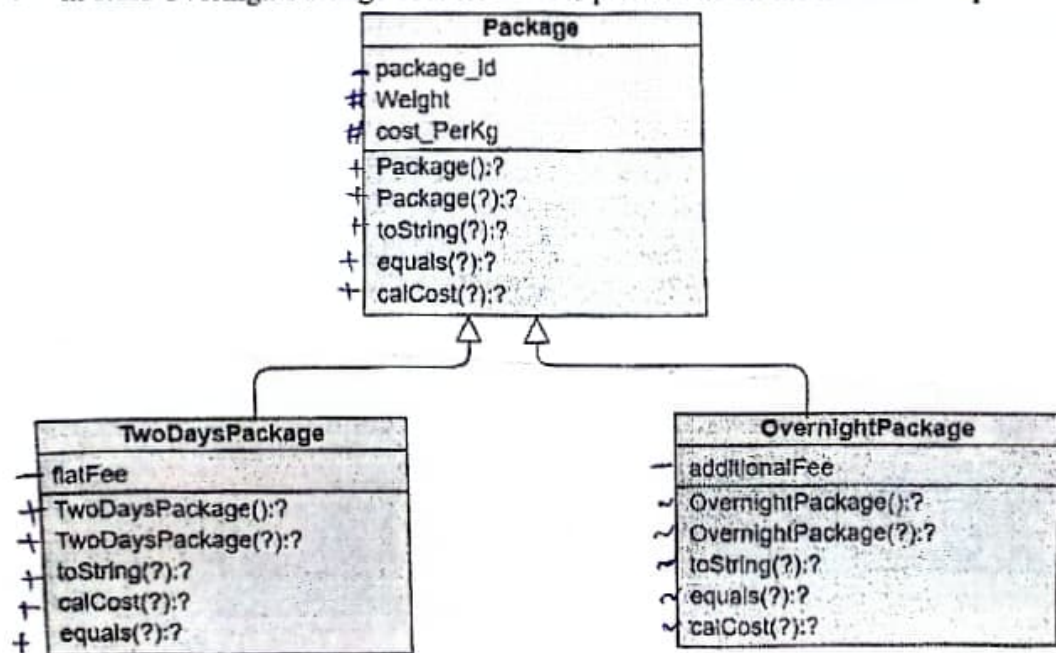
    public Address() {
        houseNo=0;
        street=null;
        city=null;
    }
    public Address(int houseNo, String street,
String city) {
        this.houseNo=houseNo;
        this.street=street;
        this.city=city;
    }
    public void setHouseNo(int houseNo) {
        this.houseNo=houseNo;
    }
    public void setStreet(String street) {
        this.street=street;
    }
    public void setCity(String city) {
        this.city=city;
    }
    public String toString() {
        return "house
no.:"+houseNo+"\nstreet:"+street+"\ncity:"+ci
ty);
    }
}
```

2. Fill the following table. [CLO-1] [Marks: 5]

| Modifier on members in class | Accessed from same class | Accessed from same package | Accessed from a different package | Accessed from a subclass in a different package |
|------------------------------|--------------------------|----------------------------|-----------------------------------|-------------------------------------------------|
| Private                      | ✓                        | ✗                          | ✗                                 | ✗                                               |
| Public                       | ✓                        | ✓                          | ✓                                 | ✓                                               |
| Protected                    | ✓                        | <del>✗</del>               |                                   | ✓                                               |
| Package protected            | ✓                        | ✓                          | ✗                                 | ✓                                               |

3. Complete the following UML notation. [CLO-2] [Marks: 5]

- In class Package package\_id is private, weight and cost\_PerKg are protected. All the methods are public methods.
- In class TwoDaysPackage flatFee is private and all the methods are public.
- In class OvernightPackage additionalFee is private and all the methods are protected.



4. Design UML notation for the following. [CLO-2] [Marks: 10]

Imagine a publishing company that markets both book and audio-cassette versions of its works. There is a class publication that stores the title and price of a publication. There are two derived classes of it; Book (data member: page count) and Tape (data member: playing time in minutes). Each of these three classes should have no argument constructor, constructor with arguments, toString() method, get and set methods for all attributes. And display function to display its data. Data members in all three classes are private while methods are public.

5. Write the java code for the following. [CLO-3] [Marks:10]

There is a class Employee with attributes SSN (String), name (String) and salary (double). It has a constructor with arguments to initialize these attributes, a readData() method, toString() method (overridden method of class Object).

There is another class CommissionEmployee that extends class Employee and has additional attributes grossSales (double), commissionRate (double). It has a constructor with arguments, readData(), toString(), and calSalary() methods.

There is a class named DailyWagesEmployee that also extends class Employee and has additional attributes hours (int) and wagesPerHour (double). It also has a constructor with arguments, readData(), toString(), and calSalary() methods.

In runner class create an array of 5 instances of type object and initialize it with different instances of CommissionEmployee and DailyWagesEmployee. Call calSalary() method for each element of the array.

6. Write the output of the following pieces of code on your answer sheet. [CLO-3][Marks: 10]

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(i) (2)</p> <pre> class A {     int i, j;     void show() {         System.out.println("i and j:" + i + " " + j);     } } class B extends A {     int k;     void show()     {         System.out.println("k: " + k);     }     void sum() {         System.out.println("i + j + k:" + (i + j + k));     } }  class Test {     public static void main(String args[])     {         B obj = new B();         obj.i = 5;         obj.j = 6;         obj.k = 9;         System.out.println("The contents are");         obj.show();         obj.sum();     } } </pre> <p>Handwritten notes: <i>ppr</i>, <i>obj [5] [6] [9]</i>, <i>1 1 k 5</i></p> | <p>(ii) (4)</p> <pre> public class Test {     public static void main(String[] args) {         m(new GraduateStudent());         m(new Student());         m(new Person());         m(new Object());     }      public static void m(Object x) {         System.out.println(x.toString());     } }  public class GoldenDelicious extends Apple { }  public class Apple extends Fruit {     public String toString() {         return "Apple";     } }  public class Fruit extends Object {     public String toString() {         return "Fruit";     } } </pre> <p>Handwritten notes: <i>D</i>, <i>A</i>, <i>F</i>, <i>O</i>, <i>Apple</i>, <i>Fruit</i></p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

(iii)

```
public class A{
    private int a=10;
    public A(){
        a*=3;
        this.show();
    }
    public void show(){
        a+=10;
        System.out.println("a="+a);
    }
}
```

a=30  
a=40

~~60~~  
~~70~~  
a=30

```
public class B extends A{
    private int b=5;
    public B(){ b.show();}
    public void show(){
        b--;
        System.out.println("b="+b);
    }
}
```

b=-1

(4)

```
public class Test {
    public static void main(String[] args) {
        A a1=new A();
        B b1=new B();
        a1.show();
        b1.show();
    }
}
```

Good Luck