Academic Year: 2023-24 **Semester: II** **Class:**

FYMCA Course Code: MC506 **Course Name: Java**

Programming Name: Durgesh Dilip Mandge
UCID : 2023510032

Experiment No. 2

Date: 05-02-2024

Aim: Fundamentals of Java Programming
CO Mapping – CO 1

Objective:

1. To create user defined package following Access control protection,
2. To develop multiple inheritance using interfaces.
3. To understand Lambda function in JAVA

Lab Exercise, Code and Output:

1.Demonstrate the significance of Package in the Program package

MyPackage1; //im

port

java.

io.*;

imp

ort

java.

util.

*;

//Factorial

public class Factorial

{

public void Factorial()

{

```
System.out.println(&quot;Enter Number To Find
Factorial:&quot;); Scanner sc = new

Scanner(System.in); int

n=s

c.ne

xtIn

t();

int

f=1;

for(int i=n;i&gt;0;i--)

{

f=f*i;

}

System.out.println(&quot;\nFactorial of &quot;+ n +&quot;

is:&quot;+f); }

}

//Maxinum

public class MaxNum

{

public void MaxNum()

{

System.out.println(&quot;\n1st

number : &quot;); Scanner sc = new

Scanner(System.in);

int mn1=sc.nextInt();

System.out.println(&quot;2nd

number : &quot;); int

mn2=sc.nextInt();
```

if(mn1>=mn2)
{

}

else

{

}

System.out.println("\nMax Num is 1st Number");

CORRECT CODE:

```
package MyPackage1;
import java.io.*;
import java.util.*;

public class Main{

 static class Factorial {
 public void findFactorial() {
 System.out.println("Enter Number To Find Factorial:");  Scanner
sc = new Scanner(System.in);
 int n = sc.nextInt();
 int f = 1;
 for (int i = n; i > 1; i--) {
 f = f * i;
 }
 System.out.println("Factorial of " + n + " is: " + f);  }
 }

 static class MaxNum {
 public void findMaxNum() {
 System.out.println("Enter 1st number:");
 Scanner sc = new Scanner(System.in);
 int mn1 = sc.nextInt();
 System.out.println("Enter 2nd number:");
 int mn2 = sc.nextInt();
 int ans;
 if (mn1 >= mn2) {
 ans = mn1;
 } else {
 ans = mn2;
 }

 System.out.println("Maximum number from " + mn1 + " and " + mn2 + " is  " + ans);
 }
 }

 public static void main(String[] args) {
 Factorial factorial = new Factorial();
 factorial.findFactorial();
```

```
 MaxNum maxNum = new MaxNum();
 maxNum.findMaxNum();
 }
}
```

OUTPUT:

```
Enter Number To Find Factorial:
8
Factorial of 8 is: 40320
Enter 1st number:
20
Enter 2nd number:
30
Maximum number from 20 and 30 is 30
```

ANSWER:

In the corrected code, the MaxNum() function is completed and all the syntax errors have been corrected. Both the functions have been defined inside a single Class.

The significance of the package declaration in the provided program lies in its ability to organize and encapsulate related classes and functionality. By specifying the package name as MyPackage1, the classes Factorial and MaxNum are grouped together within a unique namespace. This ensures that they are easily distinguishable from classes in other packages and can be accessed within the same package without explicit import statements. The package structure enhances code modularity, reusability, and maintainability by providing a logical organization to the codebase. It promotes encapsulation and information hiding, allowing classes within the package to control access to their members. Overall, the package declaration facilitates better code organization, readability, and scalability, particularly in larger projects where managing complexity is crucial.

2. Write a student admission process like student information, Roll no. allocation process, Exam process(To take marks from the user and display marks) in a package "Student". Create the object for MCA and Comp. Engg student by importing package using fully qualified way.

CODE:
StudentInfo

```
package Student;

import java.util.Scanner;

public class StudentInfo {
 private String name;
 private String department;
 private int rollNumber;

 public StudentInfo(String name, String department, int rollNumber) {
this.name = name;
 this.department = department;
 this.rollNumber = rollNumber;
 }

 public void displayInfo() {
 System.out.println("Student Name: " + name);
 System.out.println("Department: " + department);
System.out.println("Roll Number: " + rollNumber);  }
}
```

ExamProcess

```
package Student;

import java.util.Scanner;

public class ExamProcess {
 public static void takeMarks() {
 Scanner scanner = new Scanner(System.in);
 System.out.println("Enter marks:");
 int marks = scanner.nextInt();
 System.out.println("Marks obtained: " + marks);  }
}
```

MAIN

```
package lab1c;


import Student.StudentInfo;

import Student.ExamProcess;


public class Main9 {

 public static void main(String[] args) {
```

```
StudentInfo mcaStudent = new StudentInfo("Durgesh", "MCA", 101);
StudentInfo compEnggStudent = new StudentInfo("ABC", "Comp. Engg", 201);
System.out.println("MCA Student Information:");
mcaStudent.displayInfo();

System.out.println("\nComputer Engineering Student Information:");
compEnggStudent.displayInfo();

System.out.println("\nMarks for MCA Student:");
ExamProcess.takeMarks();

System.out.println("\nMarks for Computer Engineering Student:");
ExamProcess.takeMarks();
}
}
```

OUTPUT:

```
MCA Student Information:
Student Name: Dharmesh
Department: MCA
Roll Number: 101

Computer Engineering Student Information:
Student Name: ABC
Department: Comp. Engg
Roll Number: 201

Marks for MCA Student:
Enter marks:
85
Marks obtained: 85

Marks for Computer Engineering Student:
Enter marks:
80
Marks obtained: 80
```

3. Using implements runnable interface create two thread named first and second. First thread will display odd numbers from 1 to 100 and second thread will display even numbers from 1 to 100. Keep delay of 1000 ms. Show termination message when thread will exit

CODE:

```
package lab1c;

public class Main11 {
 public static void main(String[] args) {
 Thread oddThread = new Thread(new OddNumberRunnable());  Thread
evenThread = new Thread(new EvenNumberRunnable());


 oddThread.start();
 evenThread.start();
 }
}

class OddNumberRunnable implements Runnable {
 @Override
 public void run() {
 for (int i = 1; i <= 100; i += 2) {
 System.out.println("Odd Number: " + i);
 try {
 Thread.sleep(1000);
 } catch (InterruptedException e) {
 System.out.println("Odd Number Thread Interrupted");  }
 }
 System.out.println("Odd Number Thread Terminated");
 }
}

class EvenNumberRunnable implements Runnable {
 @Override
 public void run() {
 for (int i = 2; i <= 100; i += 2) {
 System.out.println("Even Number: " + i);
 try {
 Thread.sleep(1000);
 } catch (InterruptedException e) {
 System.out.println("Even Number Thread Interrupted");  }
 }
 System.out.println("Even Number Thread Terminated");
 }
}
```

OUTPUT:

OBSERVATIONS:

Through the Java exercises I learnt a breadth of topics, from basic syntax to advanced object oriented principles. Each question posed unique challenges, refining coding skills and deepening understanding of Java concepts. Through troubleshooting and experimentation, practical experience in writing efficient Java code was gained. These exercises established a solid foundation for tackling real-world programming tasks and instilled confidence for exploring complex Java projects in the future.