Dt : 10/8/2023

faq:

define "import" statement?

=>"import" is used to make class or Interface available from one package to another package

Types of importing procsses:

=>Importing process in Java can be done in three ways:

(a)Using "package_name.Class_name/Interface_name;"

(b)Using "package_name.;"*

(c)Using "Fully Qualified names"

(a)Using "package_name.Class_name/Interface_name;"

=>In this importing process the specified class_name or Interface_name is available from one package to another package.

=>This importing process is also known as "Explicit Importing Process".

Ex:

import java.util.Scanner;

import p1.EmpSalary;

Note:

=>This importing process is used when we want to make one class or Interface

available from one package to another package

(b)Using "package_name.*;"

=>In this importing process all the Class and Interfaces available from one package to another package.

=>This importing process is also known as "Implicit Importing process".

Ex:

```
import java.util.*;

import p1.*;
```

Note:

=>This importing process is used when we want to make all classes and Interfaces available from one package to another package

(c)Using "Fully Qualified names":

=>The process of declaring "classes and Interfaces" with package names part of Programming code are known as "Fully Qualified names".

EX:

```
java.util.Scanner s = new java.util.Scanner(System.in);
```

p1.EmpSalary ob = new p1.EmpSalary();

==

Assignment-1:

Convert DemoMethods5.java program into packages

ProjectName : Comparision_App

packages,

p1 : GreaterValue.java

p2 : SmallerValue.java

p3 : DemoMethods5.java(MainClass)

Assignment-2:

Convert StuMainClass.java program into packages

ProjectName : Student_App

packages,

p1 : StudentResult.java

p1 : Percentage.java

p2 : StuMainClass.java(MainClass)

==
======

Program-1(Solution)

Define a method which returns the 1 if the given number is even, in other case *return 0*

Name of method: isEven()

// which accepts an integer value as argument and

return 1 if the given number is even, else retrun 0.

Argument: int

Return type: an integer value

Example, if x = 22, return 1. if x = 35, return 0

ProjectName : Assignment1_Program_1

packages,

p1 : Test.java

```
package p1;
public class Test
{
   public int isEven(int n)
   {
        if(n%2 == 0)
        {
            return 1;
        }
        else
        {
             return 0;
        }
   }
}
```

p2 : Program1.java(MainClass)

package p2;

```
import java.util.Scanner;

import p1.Test;

public class Program1 {

        public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

    System.out.println("Enter the value:");

    int n = s.nextInt();

    Test ob = new Test();

    int res = ob.isEven(n);

    System.out.println("Result:"+res);

        }


}
```

==
==========

Program-2(Solution)

Define a method which returns the greatest number among two numbers.

Write the method with the following specifications:

Name of method: getGreatest()

// which accepts two integer values as argument and return the greatest value.

Arguments: two argument of type integer

Return type: an integer value

Specifications: The value returned by the method getGreatest() is determined by the

following rules:

If any of the given numbers are negative, return -1.

If any of the given numbers are zero, return -2.

If the given numbers are positive, return the greatest.

ProjectName : Assignment1_Program_2

packages,

p1 : Test.java

```
package p1;
public class Test
{
   public int getGreatest(int x,int y)
   {
       if(x<0 || y<0)
       {
          return -1;
       }
       else if(x==0 || y==0)
       {
           return -2;
       }
       else
       {
           if(x>y)
           {
               return x;
           }
           else
```

```
                {
                        return y;
                }
        }
    }
}
```

p2 : Program2.java(MainClass)

```
package p2;

import java.util.Scanner;

import p1.Test;

public class Program2 {

        public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

    System.out.println("Enter the value-1:");

    int v1 = s.nextInt();

    System.out.println("Enter the value-2:");

    int v2 = s.nextInt();

    Test ob = new Test();

    int res = ob.getGreatest(v1, v2);

    System.out.println("Result:"+res);

        }


}
```

===

Program-3:

Define a method which returns the least number among two numbers.

Write the method with the following specifications:

Name of method: getLeastNum()

// which accepts two integer values as argument and return the least value.

Arguments: two argument of type integer

Return type: an integer value

Specifications: The value returned by the method getLeastNum() is determined by the following rules:

If any of the given numbers are negative, return -1.

If any of the given numbers are zero, return -2.

If the given numbers are positive, return the least number.

ProjectName : Assignment1_Program_3

packages,

p1 : Test.java

```
package p1;
public class Test
{
   public int getLeastNum(int x,int y)
   {
       if(x<0 || y<0)
       {
          return -1;
```

```
        }
        else if(x==0 || y==0)
        {
            return -2;
        }
        else
        {
            if(x<y)
            {
                return x;
            }
            else
            {
                return y;
            }
        }
    }
}
```

p2 : Program3.java(MainClass)

```
package p2;

import java.util.Scanner;

import p1.Test;

public class Program3 {

        public static void main(String[] args) {

      Scanner s = new Scanner(System.in);

      System.out.println("Enter the value-1:");

      int v1 = s.nextInt();

      System.out.println("Enter the value-2:");

      int v2 = s.nextInt();
```

```
        Test ob = new Test();

        int res = ob.getLeastNum(v1, v2);

        System.out.println("Result:"+res);

          }


}
```

==
=========

Program-4:(Solution)

Define a method which returns the number it if it is an even number,

if the number is odd then return the next multiple of 10.

Write the method with the following specifications:

Name of method: oddRounder()

// which accepts an integer value as argument and return the same value

if it is an even number, if the value is odd then return the next multiple of 10.

Arguments: one argument of type integer

Return Type: an integer value

Example if x = 24 then return 24, if x = 25 then return 30.

Specifications: The value returned by the method oddRounder() is determined by the following rules:

If any of the given number is negative, return -1.

If any of the given number is zero, return -2.

If the given number is even, return the same number.

If the given number is odd, return the next multiple of 10.

ProjectName : Assignment1_Program_4

packages,

p1 : Test.java

```java
package p1;
public class Test
{
    public int oddRounder(int n)
    {
        if(n<0)
        {
            return -1;
        }
        else if(n==0)
        {
            return -2;
        }
        else if(n%2 == 0)
        {
            return n;
        }
        else
        {
            int q = n/10;
            return (q+1)*10;
        }
    }
}
```

p2 : Program4.java(MainClass)

package p2;

```
import java.util.Scanner;

import p1.Test;

public class Program4 {

        public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

    System.out.println("Enter the value:");

    int n = s.nextInt();

    Test ob = new Test();

    int res = ob.oddRounder(n);

    System.out.println("Result:"+res);

        }

}

========================================================================
```