Dt : 5/8/2023

Ex-program : DemoBlock3.java

```
class BTest2
{
        int a=10;
        static int b=20;

        {
                a++;
                b++;
                System.out.println("****Instance-block****");
                System.out.println("The value a:"+a);
                System.out.println("The value b:"+b);
        }
}
class DemoBlock3
{
        public static void main(String[] args)
        {
                System.out.println("=====ob1=====");
                BTest2 ob1 = new BTest2();
                System.out.println("=====ob2=====");
```

```
            BTest2 ob2 = new BTest2();
        }
}
```

o/p:

```
=====ob1=====
****Instance-block****
The value a:11
The value b:21
=====ob2=====
****Instance-block****
The value a:11
The value b:22
```

Execution flow of above program:

ClassFiles:

BTest2.class

DemoBlock3.class(MainClass)

==========

faq:

wt is the diff b/w

(i)Methods

(ii)Blocks

=>Methods are executed on method_call,but blocks are executed automatically without

calling.

=>Blocks will have highest priority in execution than methods.

=>Static blocks will have highest priority in execution than static methods.

=>Instance blocks will have highest priority in execution than Instance methods.

==========

faq:

wt is the diff b/w

(i)Instance block

(ii)Constructor

=>Both components are executed while Object creation process,but Instance block

will have highest priority in execution than Constructor.

==========

faq:

wt is the diff b/w

(i)static block

(ii)Constructor

=>staic block is one time executable component at class_level.

=>Constructor also one time executable component at Object_level.

==========

faq:

wt is the execution behaviour of Constructor declared with return_type?

=>If any constructor declared with return_type then its behaviour is normal Instance method.

==========

Ex-program : DemoCon6.java

```
class BTest3
{
	static
	{
		System.out.println("****Static block****");
	}
	BTest3()
	{
		System.out.println("****Constructor*****");

	}
```

```
        {
                System.out.println("****Instance-Block****");
        }
}
class DemoCon6
{
        public static void main(String[] args)
        {
                BTest3 ob = new BTest3();

        }
}
```

o/p:

```
****Static block****
****Instance-Block****
****Constructor*****
```

==========

Dt: 8/8/2023

**imp*

Packages in Java:

=>Package is a collection of "Classes and Interfaces"

=>we use "package" keyword to declare packages

=>The packages in Java are categorized into two types:

1.Pre-defined packages

2.User defined packages

1.Pre-defined packages:

=>The packages which are defined and available from JavaLib are known as pre-defined packages or Built-in packages.

=>Thw following are some important pre-defined packages:

lang - Language package

util - Utility package

io - Input/Output Stream package

net - Networking package

2.User defined packages:

=>The packages which are defined by the programmer are known as User defined packages.

**imp*

Creating JavaProject with packages using IDE Eclipse:

(IDE - Integrated Development Environment)

step-1 : Download IDE Elipse from the following link and Install

https://www.eclipse.org/downloads/.../eclipse-ide-java-developers

Note:

select - Eclipse Ide for Enterprise Java and Web Developers

step-2 : Open IDE Ecplipse,while opening name the WorkSpace(working folder) and

click "Launch"

step-3 : Create Java Project

click on File->New->Project->Java->Select "Java Project" and click "Next"->

name the project and click "Finish"

step-4 : Create packages in "src"

RightClick on "src"->new->package,name the package and click "Finish".

step-5 : Create classes in Packages

RightClick on packages->new->Class,name the Class and click "Finish"

Note:

=>To increase font,click on Window->Preferences->General->Appearance->

Colors and Fonts->Java->Java Editor Text Font->...

ProjectName : EmployeeSalary_App

packages,

p1 : EmployeeSalary.java

```
package p1;
public class EmpSalary {
    public float cal(int bSal) {
        float tSal = bSal+(0.93F*bSal)+(0.63F*bSal);
        return tSal;
    }
}
```

p2 : DemoEmployee.java(MainClass)

```
package p2;

import java.util.Scanner;

import p1.EmpSalary;

public class DemoEmployee {

    public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

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

    int bSal = s.nextInt();

    EmpSalary ob = new EmpSalary();
```

```
    float totSal = ob.cal(bSal);
    System.out.println("TotSal:"+totSal);
        }
}
```

step-6 : Execute the program

open MainClass,Click Run->Run

==========

Diagram:

==========