What is Eclipse IDE in java ?

IDE stands for "Integrated Development Environment". By using eclipse IDE, we can develop, compile and execute our java program in a single window.

The main purpose of Eclipse IDE is reduce the development time, since the development time will be reduced then automatically the cost of the project will be reduced.

How to create a Project in Eclipse IDE :

File -> new -> Project OR Java Project -> Provide the name for the project (Watch 53) -> Finish

What is a Package in java :

* A package is nothing but folder in window O.S.

Store the data in the following two approaches :

Approach 1 :

The two major drawbacks in Approach 1 :
----
a) Fast searching is not possible
b) Name cannot be re-usable (We cannot re-use the name)

Approach 2 :

In this approaching 2, Fast searching is possible as well as Name can be re-usable

A package is nothing but folder in windows. It is used to arrange the classes and interfaces into a particular group.

If we arrange our java classes into a particular group by using packages (folders) then we will get the following two advantages :

1) Fast searching is possible.
2) Name can be reusable.

Types of Packages :
----
1) Predefined OR Built-in package : The packages which are created by java software people for arranging the programs are called predefined package.

Example : java.lang, java.util, java.io, java.sql, java.net and so on

2) Userdefined Package OR Custom package : The packages which are created by user for arranging the user-defined programs are called user-defined package.

Example :
basic;
com.ravi.basic;
com.tcs.online_shopping;

==
What is a class in Java : [Introduction]

It is a keyword to define fields and methods (static and non static both).

As we know from JDK 25V, based on our file name, java compiler will automatically generate class.

If a user explicitly defines a class then java compiler will not generate any class. It will depend upon the class which is written by user.

Is a single .java file, How many public classes we can write ?

* Only one public class we can write and it must be our file name.
Example :
Test.java   [File name is Test.java so Only Test class we can declare as public]
public class Test
{
}

* In a single java file we must have only one public class but it **may contain n number of non public classes.**
Test.java
```
public class Test      //Only Test class we can declare as public
{
}

class Demo
{
}

class Welcome
{
}
```
In this approach, total 3 .class files are created. Java compiler will generate the .class file for all the classes whether it is public OR non public

WAP to show method overloading by using ECLIPSE IDE :
package com.ravi.method_overloading;

```
public class MethodOverloading
{
   void main()
   {
      IO.println("Sum of two Integer is :"+this.accept(12,90));
      IO.println("Sum of three Integer is :"+this.accept(12,90, 10));
   }

   int accept(int x, int y)
   {
      return x + y;
   }

   int accept(int x, int y, int z)
   {
      return (x+y+z);
   }
}
```

What is BLC and ELC class in java ?

BLC :

BLC stands for Business Logic class. In this class we are responsible to write the logic. This class will not contain main method.
The main purpose of this BLC class to reuse this class in various packages.

Example :
//BLC
public class Calculate
{
   //Here we are responsible to write the logic
}

ELC :

It stands for Executable Logic class. It will not contain any logic but the execution of the program will start from this ELC class because it contains main method.

Example :
//ELC
public class Main
{
   void main()
   {
   }
}

==
Working with static methods with various return type :

* A static method must be declare with static modifier.
* A static method can be call with the help of class name here Object is not required.

Square.java (BLC class)
----
package com.ravi.blc;

public class Square
{
   public static void getAreaOfSquare(int side)
   {
      IO.println("The Area of Square is :"+(side*side));
   }
}

Rectangle.java(BLC class)
----
public class Rectangle
{
   public static void getAreaOfRectangle(int length, int breadth)
   {
      IO.println("Area of Rectangle is :"+(length * breadth));
   }
}

Main.java (ELC class)
----
package com.ravi.elc;

import com.ravi.blc.*;

public class Main
{
   void main()
   {
      Square.getAreaOfSquare(5);
      Rectangle.getAreaOfRectangle(2, 5);
   }
}
```