Access Protection in Java

There are four access protections or modifiers in java
Default, public, private and protected.

For class:

Modifiers	Allowed/ Not Allowed
Default	Allowed [Package level access only without import]
public	Allowed [Outside package can be accessible with import]
private	Not allowed
Protected	Not allowed

For variable:

Member Type	Inside its own class	Inside Same Package		Inside another Package	
		Sub class	Another Class	Sub class	Another Class
default	Y	Y	Y	N	N
private	Y	N	N	N	N
public	Y	Y	Y	Y	Y
protected	Y	Y	Y	Y [Non-static]	N

```
public class ChildClass{
}

class ChildClass{
}
```

Both default and public modifiers are allowed for class as making class private and protected would restrict other classes to use inheritance properly. If a class is marked private then this class can't be used in inheritance and can't even make an object of this class. If a class is marked protected then it would be a redundant feature of java as the same feature is provided by the abstract classes in java.

Private variable in Java

When a variable is marked private it means it is designed for its own class and can't be accessible through object creation or inheritance.

Default variable in Java

When a variable is marked default it means it is designed for its own package and can't be accessible outside its own package

```
package unit_01

public class VariablesInJava{
        int a = 10; //default protection
        public int b = 10;
        public static void main(String[] args){
                System.out.println(a);
        }
}
```

```
package unit_02

import unit_01.VariablesInJava;

public class VariablesInJava2{
        public static void main(String[] args){
                VariablesInJava obj = new VariablesInJava();
                System.out.println(obj.a); // Not visible
                System.out.println(obj.b); // Visible
        }
}
```

As we are trying to access variable "a" outside its own package then it will not be visible. Only public data in this case "b" will be visible in another package.

Public variable in Java

When a variable is marked public it means it is designed for its own class, same package and another package as well. It is accessible anywhere through object within the same project.

Protected variable in Java

When a variable is marked protected it means it can be accessible anywhere throughout project using inheritance or sub class except in one situations when you try to access a protected variable in another package and inside a non-sub class.

```
package unit_01
public class VariablesInJava{
        protected int a = 10;
         public int b = 10;
        public static void main(String[] args){
                System.out.println(a);
        }
}
```

```
package unit_02
import unit_01.VariablesInJava;

public class VariablesInJava2 extends VariablesInJava {
        public static void main(String[] args){

                VariablesInJava2 obj = new VariablesInJava2();
                obj.accessProtected();

        }
       public void accessProtected() {
                System.out.println(this.f);
                this.display();
        }
}
```

Package and class Import in Java

With the help of import statement, we can access public class even outside the package and if the class is default then it can be accessed through object creation inside the same packages anywhere.

```
package unit_01
public class VariablesInJava{
        protected int a = 10;
         public int b = 10;
        public static void main(String[] args){
                System.out.println(a);
        }
}
```

```
package unit_01
class VariablesInJava2{
        public static void main(String[] args){
                VariablesInJava obj = new VariablesInJava();
                System.out.println(obj.b);
        }
}
```

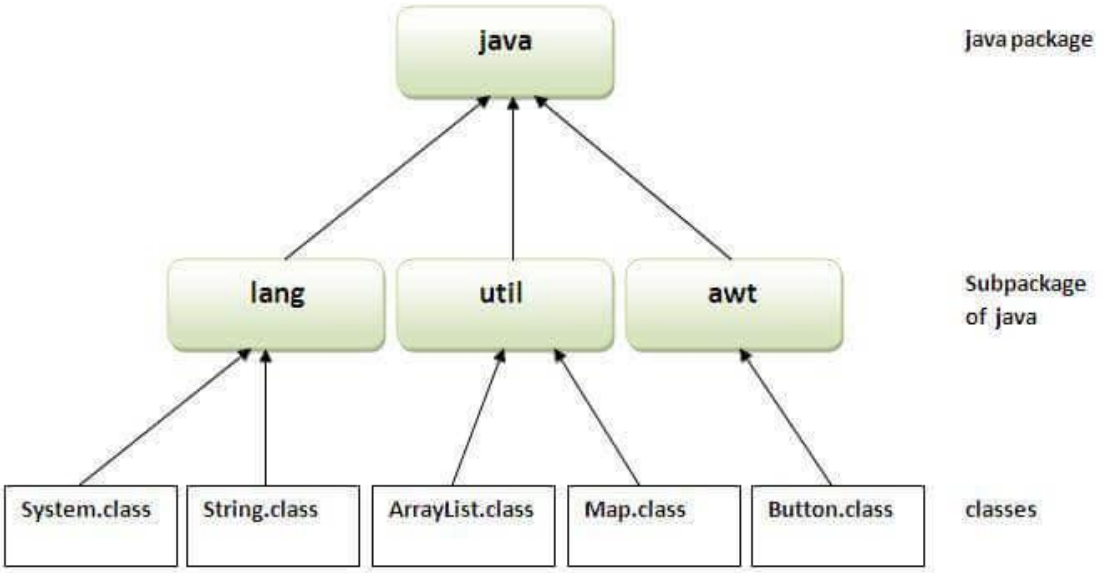
```
package unit_02

//only when VariablesInJava is a public class
import unit_01.VariablesInJava2; // VariablesInJava2 is a default class and
can't be seen outside the package

public class VariablesInJava3 {

}
```

Java Package Hierarchy



There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

- lang provides classes that are fundamental to the design of the Java programming language such as System.class and String.class.
- util provides classes like ArrayList.class and Map.class as collection framework in Java. **[Unit-04].**
- awt provides classes to design front end interface in java like Button.class. **[Unit-04].**

Static Import in Java

> Static members of a class can be accessed without class name or any object.
> For Example: we always use sqrt() method of Math class by using Math class i.e. Math.sqrt(), but by using static import we can access sqrt() method directly.
> out.println() can be used instead of System.out.println() as "out" is the static data in System Class

```
import static java.lang.System.*;
import java.lang.Math;

public class P11_Task02_ImportingAndPackagesInJava {
	public static void main(String args[]) {

		out.println("Welcome to package");
		out.println(Math.sqrt(4));
		out.println(Math.pow(2,2));
		out.println(Math.abs(6.3));

	}
}
```