SYNTAX TECHNOLOGIES

JAVA

Class 18

Agenda

Methods with return values and without

Access Modifiers

Practice on: methods, access modifiers

Task

1. Create a method that will take 3 parameters as a numbers and returns the largest

2. Create a method that will take 1 parameter as a String and return reversed String.

3. Create a method that will take 1 parameter as a String and return whether String is palindrome or not.

Access Modifiers

Access modifiers in Java help to restrict the scope of a class, constructor , variable , method or data member.

These are used to control access to data members or methods.

In java programming these are classified into four types:

private = native class ,
default = package
protected = inherited
public = universal

Access Modifiers

Access modifiers are always used to control the reuse of features within a package and access between classes , interfaces and interface to class.

Access modifiers provides features accessing and controlling mechanism among classes and interfaces

Note:
Default is not a keyword (like public, private, protected are keyword)

Access Modifiers

Visibility	Public	Private	Protected	Default
Within Same Class	Yes	Yes	Yes	Yes
From Any Class in Same Package	Yes	No	Yes	Yes
From Any Subclass in Same Package	Yes	No	Yes	Yes
From Any Sub Class from Different Package	Yes	No	Yes(**Only By Inheritance**)	No
From Any Non Subclass in Different Package	Yes	No	No	No

Default

If we are not using private , protected and public keywords then JVM is by default taking as default access modifiers.

The scope of this modifier is limited to the package only. This means that if we have a class with the default access modifier in a package, only those classes that are in this package can access this class. No other class outside this package can access this class.

Similarly, if we have a default method or data member in a class, it would not be visible in the class of another package.

The data members, class or methods which are not declared using any access modifiers i.e. having default access modifier is accessible to the classes which are defined in the same package.

Example of default access modifier

```
package abcpackage;
public class Addition {
   int addTwoNumbers(int a, int b){
      return a+b;
   }
}
```

```
package xyzpackage;
import abcpackage.*;
public class Test {
   public static void main(String args[]){
      Addition obj = new Addition();
      obj.addTwoNumbers(10, 21);
   }
}
```

OUTPUT

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The method addTwoNumbers(int, int) from the type Addition is not visible
at xyzpackage.Test.main(Test.java:12)
```

Public

if a variable is set to the public it can be accessed from any class available in the Java world.
Any Method in any Class can access the given variable via Inheritance or Direct access.

Example of Public access modifier

```
package abcpackage;
public class Addition {
   public int addTwoNumbers(int a, int b){
      return a+b;
   }
}
```

```
package xyzpackage;
import abcpackage.*;
class Test{
   public static void main(String args[]){
      Addition obj = new Addition();
      System.out.println(obj.addTwoNumbers(100, 1));
   }
}
```

Output: 101

Protected

If a variable is set to protected inside a Class, it will be accessible from its subclasses defined in the same or different package **only via Inheritance.**

Protected data member and method are only accessible by the classes of the **same package and the subclasses present in any package.**

You can also say that the protected access modifier is similar to default access modifier with one exception that it has visibility in subclasses.

This access modifier is generally used in a parent child relationship.

The methods or data members declared as protected are **accessible within same package or sub classes in the different package.**

The protected access modifier can be applied on the data member, method and constructor. **It can't be applied on the class.**

Example of Protected access modifier

```
package abcpackage;
public class Addition {
   protected int addTwoNumbers(int a, int b){
       return a+b;
   }
}
```

```
package xyzpackage;
import abcpackage.*;
class Test extends Addition{
   public static void main(String args[]){
       Test obj = new Test();
       System.out.println(obj.addTwoNumbers(11, 22));
   }
}
```

Output: 33

Private

A variable defined private will be accessible only from within the Class in which it is defined. Such variables are not accessible from outside the defined Class, not even in its subclass.

Private Data members and methods are only accessible within the class

Class and Interface cannot be declared as private

If a class has private constructor then you cannot create the object of that class from outside of the class.

Any other **class of the same package will not be able to access** these members.

Example of Private access modifier

```
class ABC{
   private double num = 100;
   private int square(int a){
         return a*a;
   }
}
public class Example{
   public static void main(String args[]){
         ABC obj = new ABC();
         System.out.println(obj.num);
         System.out.println(obj.square(10));
   }
}
```

Compile - time error