11-02-2013 | MONDAY

execute command: java packagename.javafilename

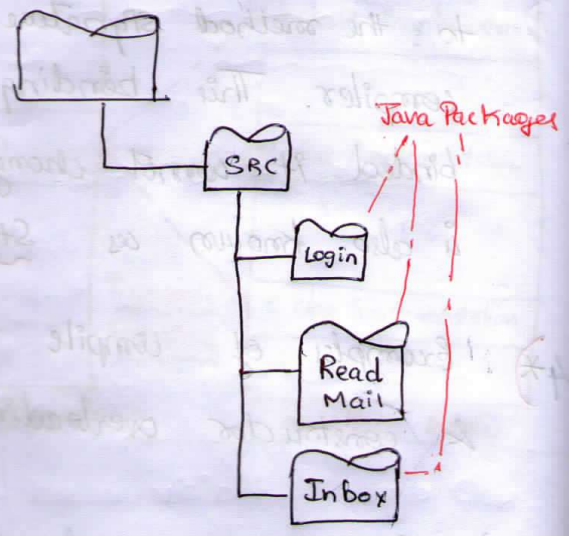
PACKAGES

P88

1)
```
package login; // package declaration

class A
{
  psvm (String[] args)
  {
    Sop("running class A from login");
  }
}

class B
{
  psvm (String[] args)
  {
    Sop ("running class B from login");
  }
}
```

O/p:

2)
```
package inbox; // package declaration

class C
{
  psvm (String[] args)
  {
    Sop("running class C from inbox");
  }
}

class D
{
  psvm (String[] args)
  {
    Sop ("running class D from inbox");
  }
}
```



1* While developing java classes the java classes are grouped into packages.

2* Packages are developed mainly to organise java classes and for reusability purpose.

* ~~Referring a jar class name~~

3* A class associated with a package should be executed by using fully qualified class name

4* Referring a java class along with package name is known as Fully qualified class

5* A class of one package can be referred from other package either by fully qualified class name (or) by importing the package.

3) P89

```
package login;
class A
{
    int i=120;          // access privelage : default
    psvm(String[] args)
    {
      Sop("Program starts...");
      A a1 = new A();
      Sop("i="+a1.i);
      Sop("Program ends...");
    }
}

class B
{
    psvm(String[] args)
    {
      Sop("Program starts...");
      A a1=new A();
      a1.i=234;
      Sop("i="+a1.i);
      Sop("Program ends...");
    }
}
```

4) P90

```
A* private int i=120;  //access privalage: private
```

O/p: class B
error

6* Import statement should be written after package declaration statement

7* We can import one class at a time (or) all classes of the package.

5) P91

```
package login;
public class A
{
    public int i=120;
    psvm ( )
    {
      Sop("program starts...");
      A al = new A();
      Sop ("i = "+al.i);
      Sop ("Program ends...");
    }
}
```

```
package inbox;
public class C
{
    psvm ( )
    {
      Sop (" Program starts...");
      login.A al = new login.A();
      al.i = 785;
      Sop ("i ="+al.i);
      Sop ("Program ends...");
    }
}
```

Access Specifications

1* The access specifiers defines visibility

2* Java supports 4 types
1) Private
2) default
3) protected
4) Public

3* Private: has a visibility upto class level members
They cannot be referred (or) accessed outside the class.

4* default members: has a visibility upto package level. The default members can be referred from any class within the package.

5* public members has a wider visibility. Public members of an class can be referred from outside the package or within the package.

Public members of an class can be referred from outside package only if the class is public.

6* Protected members has visibility upto package level, it can be referred outside package through inheritance, once a protected members are inherited behaves like a private in subcla

12-02-2013 | Tuesday

P92

6) Same as previous example
package login;
public class A

```
package inbox;
import login.A;
public class C extends A
{   psvm( )
  {  Sop("Program starts...");
     C c1 = new C();
     c1.i = 7665;
     Sop("i= "+c1.i);
     Sop("Program ends...");
  }
}
class D extends C
{
   psvm(String[] args)
   {
     Sop("Program starts...");
     D d1 = new D();
      d1.i = 5465;
     Sop("i= "+d1.i);
     Sop("Program ends...");
   }
}
```

7) An outer class should be either public or default

8) Inner class can have any of the four access specifiers.

9) An outer class cannot be declared as private bec it cannot be used or referred anywhere.

10) An outer class cannot be declared as protected bec protected members can be referred only through inheritance, class will not be inherited, only members of class will be inherited.

11) A javafile can have only one public class, in such case the filename should be same as public class.

12) A private variable of a class can be referred from another class by using getters & setters method. Both getters & setters method should be public.

7)

```
package login;
class C
{
  private C()
   {
    Sop("running C() constructor");
   }
  psvm(String[] args)
  {
   Sop ("Program starts...");
   C c1 = new C();
   Sop("Program ends...");
  }
}
class Run1
{
   psvm (String[] args)
   {
    Sop("program starts...");
    C c1 = new C();
  // error, Cant refer private constructor from another class
    Sop("Program ends...");
   }
}
```

8)

* A java class which has following is known as Java Bean class

1) public constructors
2) private variables
3) public getters & setters method.

* java bean are used in development of enterprise application

Getters Method & Setters Method

8) P94

```
package login;
class Student
{
   private int stID=191;
   private double stMarks = 12.58;
   // getters Method
   public int getStID()
   {
      return stID;
   }
   public double getStMarks()
   {
      return stMarks;
   }
   // setters Method
   public void setStID(int stID)
   {
      this.stID = stID;
   }
   public void setStMarks(double stMarks)
   {
      this.stMarks = stMarks;
   }
}
class Run2
{  psvm (String[] args)
   {
      Sop ("Program starts...");
      Student st1 = new Student();
      Sop ("Student ID: " + st1.getStID());
      Sop ("Student Marks: " + st1.getStMarks());
      Sop ("Program ends...");
   }
}
```

Can u refer a private member of a Class?

Yes, by getters, setters method() and set the methods as public.

* Getters method returns private variable value.
* Setters method initialize/ sets the private variable value.
* ~~#~~ Gossa

O/p: Program starts...
Student ID: 191
Student Marks: 12.58
Program ends...