JavaNotes

****

Prepared by Uday Pawar

**Java Libraries**

**1. It is a collection of pre-defined packages.**

**2. Each package/folder is collection of pre-defined classes and pre-defined interfaces.**

**3. Each class or interface is a collection of variables and methods.**

**4. All the pre-defined classes and interfaces are present inside a jar file called as rt.jar (rt->runtime) or zip file -> src.zip**

**Write Any 6 pre-defined package Names**

**1.java.lang**

**2.java.util**

**3.java.io**

**4.java.sql**

**5.java.applet**

**6.java.awt ..........**

**---------------------------------------**

**1. java.lang package**

**- This package is implicitly or automatically imported in all java classes and interfaces.**

**2. Object Class**

**- Object is a pre-defined class present in java.lang package.**

**- Object class is referred as super-most class in java.**

**- Object class is implicitly inherited by all java classes.**

**methods present in Object Class**

**1. protected Object clone()**

**2. public boolean equals(Object o)**

**3. public int hashCode()**

**4. public String toString()**

**5. public void wait()**

**6. public void wait(long a)**

**7. public void wait(long a, int b)**

**8. public void notify()**

**9. public void notifyAll()**

**10. public Class getClass()**

**11. protected void finalize()**

**----------------------------------------**

**1. toString()**

**syntax : public String toString()**

**- toString() returns the String representation of an Object in the below format.**

**FullyQualifiedClassName @ HexadecimalValueOfTheHashCode**

**Programs:**

**1.**

**package com;**

**import java.util.Scanner;**

**import java.util.ArrayList;**

**public class Demo**

**{**

**Scanner s = new Scanner(System.in);**

**ArrayList l = new ArrayList();**

**Thread t = new Thread();**

**Object o = new Object();**

**}**

**2.**

**package** com;

**public** **class** Car {

**public** **static** **void** main(String[] args) {

// Up-casting

Object obj = **new** Car();

Car c = **new** Car();

}

}

**3. Without Overriding toString()**

**package** com;

**public** **class** Person

{

**public** **static** **void** main(String[] args)

{

Person p = **new** Person();

System.***out***.println(p); // implicitly it calls toString() -> String Representation

System.***out***.println(p.toString()); // explicitly calling toString()

}

}

**o/p:**

com.Person@15db9742

com.Person@15db9742

**4. After overriding toString()**

**package** org;

**public** **class** Person

{

@Override

**public** String toString()

{

**return** "Hi Guldu!";

}

**public** **static** **void** main(String[] args)

{

Person p = **new** Person();

System.***out***.println(p); // implicitly calls toString()

System.***out***.println(p.toString()); // explicitly calls toString()

}

}

**o/p:**

Hi Guldu!

Hi Guldu!

**5.**

**package** org;

**public** **class** Student

{

String name;

Student(String name) {

**this**.name = name;

}

@Override

**public** String toString() {

**return** "Name: "+name;

}

**public** **static** **void** main(String[] args)

{

Student s1 = **new** Student("Tom");

Student s2 = **new** Student("Jerry");

System.***out***.println(s1);

System.***out***.println(s2);

System.***out***.println(s1.toString());

System.***out***.println(s2.toString());

}

}

o/p:

Name: Tom

Name: Jerry

Name: Tom

Name: Jerry

**6.**

**package** org;

**public** **class** Employee {

**int** id;

String name;

Employee(**int** id, String name) {

**this**.id = id;

**this**.name = name;

}

@Override

**public** String toString() {

**return** "Employee Id of "+name+" is "+id;

}

**public** **static** **void** main(String[] args) {

Employee e1 = **new** Employee(101, "Ambani");

Employee e2 = **new** Employee(102, "Tata");

Employee e3 = **new** Employee(103, "Cook");

System.***out***.println(e1);

System.***out***.println(e2);

System.***out***.println(e3);

}

}

o/p:

Employee Id of Ambani is 101

Employee Id of Tata is 102

Employee Id of Cook is 103