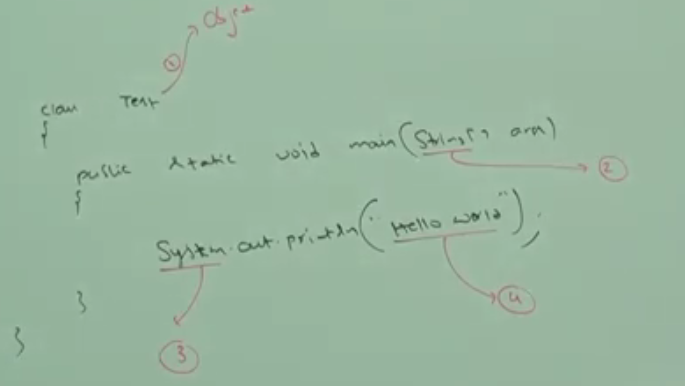
Introduction

1. To write any program, the most common classes and interfaces are grouped together into a separate package in java.lang package.
2. Without using util, sql packages, we can use java program.
3. Let’s write a program:  
     
   NOTE: Without using lang package, we can’t use keyword **class** as every class inherits from Object class which is in java.lang package.

in this snapshot, all red numbers belong to java.lang package.

NOTE: Without using lang package, we can’t write even a very simple minimum program, so java.lang package is imported automatically.

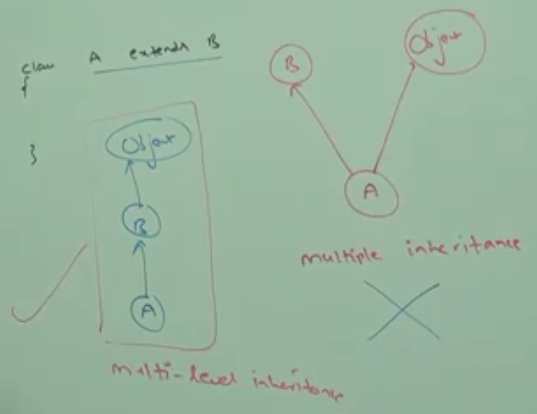
1. d

Object(C)

1. Why is Object(c) class is the top root class?   
   **Answer**: There are some methods which is applicable to any object of any class (pre-defined or user-defined/customized class). Those methods are as follows:

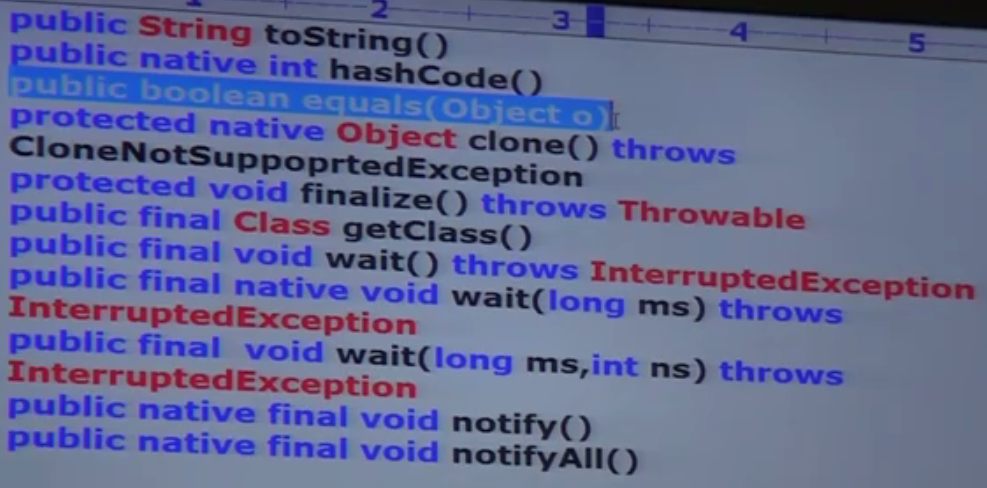
* hashCode()
* equals()

We put them into a separate class called **Object**.

1. Why not is the String class the top root class of very other classes?  
   Answer: String Class contains methods we can’t be applied to any object of other classes. Such as String class contains length() which is not applicable to Student class object.
2. Every class in java is either directly or indirectly child class of Object Class.
3. Object class is considered as **Root** class of every other class.
4. If our class doesn’t extend any other class, then only our class is the direct child class of **Object class**.  
   If our class extends any other class, then our class is indirect child class of **Object class**.  
   
5. d

Object class Methods

1. Object Class defines the following 11 methods

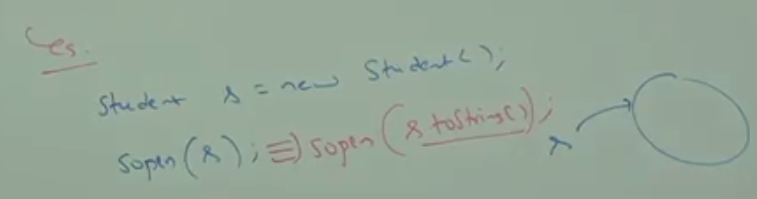
Strictly speaking Object class contains 12 methods. The extra method is **registerNatives() which is** internally used by Object class itself. Hence we are not required to consider this method.

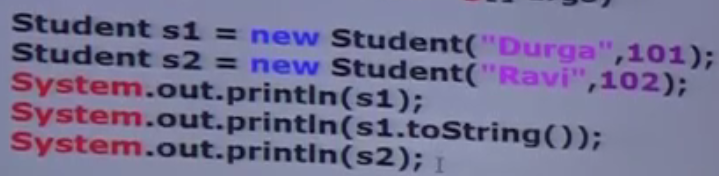
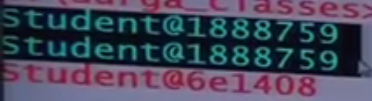
1. d

Object Class methods Description

# toString()

We can use toString() method, to get the string representation of an object.  
String s = obj.toString();  
Whenever we are trying to print object reference, internally toString() method will be called.  
Sysout.out.println(student1) 🡺 System.out.println(student1.toString());



if our class doesn’t contain toString(), then Object class toString() will be executed.  
 

toString() in Object class  


**className@hashCode\_in\_hex\_form**

Not meaningful output.

So override toString() method in your class to get meaningful output.  
In all Wrapper classes, toString() method is already overridden to return the content of wrapper class instance.  
All Collection classes’ toString() method is also overridden.

