OOPS concepts

Abstraction

Primitive Data Types:byte, short,int,long,float,double,char, boolean

Access modifiier

public --u can acccess any where and everywhere

private-u can access methods, data members within class

protected-u can access all class within package and outside subclass in the package

default-u can access all class within same package

Strings, String buffer, String builder

= = method and Equals() method

== compares two reference objects(string), its is fastest one

Equals() method compares two values of object(string)

String s1= new String( “gomathy”); s1==s2 false

S1.equals(s2) true

String s2= new String(“gomathy”);

Constructor- is use to initiaialize the object.Constructor name must be same as its class name. It is invoked at the time of object creation and used to initialize the state of an object. don’t have any return type .default constructors and parameter constructors

Exception handling

Static keyword,-it is use to represent class members.it can be used variable,method,nested class.

Static is class level. There is no need to create an object to invoke in method ,variable , instead of we can call class name . Static method cannot call non-static method. Mainly saves memory management

super keyword-refers immediate super class of an object

this keyword- refers to the current instance of the class.

final keyword-is very restricted class, variables and methods, class

Wrapper class

Primitive data types

Auto boxing and unboxing

Type casting---converting data from one data type to another data type

implicitly casting (up casting)widening(byte-short-int-long-float-double)converting small data type to big size data type

Explicit casting (down casting)(double-float-long-int-short-byte) –higher range to small range

Garbage collector

Comparator and comparable

Equals () and hash code

If you are overriding equals method then you should override hashcode() also.

If two objects are equal then they must have same hashcode.

If two objects have same hashcode then they may or may not be equal

Always use same attributes to generate equals and hashcode as in our case we have used name.

Here are two rules that are good to know about implementing the hashCode() method in your own classes, if the hashtables in the Java Collections API are to work correctly:

1. If object1 and object2 are equal according to their equals() method, they must also have the same hash code.
2. If object1 and object2 have the same hash code, they do NOT have to be equal too.

In shorter words:

1. If equal, then same hash codes too.
2. Same hash codes no guarantee of being equal.

If the two obj are equal then only hash code will enter into bucket and look on value are equals.

local variable-inside method, outside the class

instance variable--- inside the class

static variable

Enumeration - It provides the methods to get a series of elements from a collection of object taking one at a time… data type which contains fixed set of constants. use for days of week, month of year, direction etc...

java Jdbc connection

signature-method name,type of parameters

java i spass by value, not pass by refererence, eg in cannot chage original value but it can change local variable or instance variable.

Collection-group of objects into single unit. Collections---- it is use to store,manipulate and retrive data

interface,implemtaions,algorthims

Interface, implements class,extends.

Interface

1.list

2.Set

3.map

4.Queue

5.sorted map

------

List

1.Arraylist ------class -----extends

2.vector --class

3.Linkedlist-----classs

Set

1.HashSet -classs

2.Treeset

3.linkedHashset

Map

1.Hashmap -class

2.HashTable

3.Treemap

LinkedHashMap

Queue

1.priority queue

In a java program execution memory is divided into three parts:

1. 1. Stack.
2. 2. Heap.
3. 3. Class/Method area.
4. ***1. Stack: Stack is used to local variables of the methods.***
5. ***2. Heap: Heap is used to store objects.***
6. ***3. Class Area: Class area is used to store static data members.***

Static:

Static is a keyword in java used to represent the class members. It can be used with variable, method, initializer block and nested class.

## Limitations of static methods and static initialize blocks.

1. **1. Non-static data members can’t be accessed in static methods and static initialize blocks.**
2. **2. Non-static methods can’t be invoked in static methods and static initialize blocks.**
3. **3. This or super keyword can’t be refers in static methods and static initialize blocks.**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **final** | **Finally** | **finalize** |
| 1) | Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed. | Finally is used to place important code, it will be executed whether exception is handled or not. | Finalize is used to perform clean up processing just before object is garbage collected. |
| 2) | Final is a keyword. | Finally is a block. | Finalize is a method. |

Method overloading in java

Method overloading means more than one methods in a class with same name but different parameters. (same methods but differernt parmaters)

Compile time polymorphism or static binding

Compiler resolve method call by matching method signature(methods and parameters) at compile time, that’s why it is known as static .

method overloading is not possible by changing return type of the method .

Method overriding in java (run time polymorphism)

Same methods and same parameters

If a subclass provides a method with the same signature (name and parameter) as in its super class, then subclass overrides the method of its super class. This process of overriding a super class method by subclass is known as method overriding.

Conditions for method overriding: 1. Method in subclass must have same signature as in its super class. 2. Two classes must follow IS-A relationship. –

## Can static method be overridden?

No, Static methods can’t be overridden because they are associated with class not with the object.

## Role of access modifiers in method overriding:

Access modifier of overridden method in subclass can’t be more restrictive than in super class. Otherwise it will throw an exception.

 You can not override static, final and private method in java

super keyword in java

super is a keyword in java which refers to the immediate super class object.

Use of super keyword in java:  1. super can be used to call immediate super class constructor (constructor chaining). 2. super can be used to call immediate super class method on a subclass object from a subclass method. 3. super can be used to access immediate super class instance variable.

This keyword:

Refers the current instance of the class. Every constructor and all non-static methods in java have this as an implicitly parameter.

Public classexample{

Public void classexample(int a,int b){

This.a =a;

This.b =b;

}

## Use of[*this keyword in java*](http://tutorialspointexamples.com/this-keyword-in-java/):

1. 1. this can be used to differentiate between instance variable and local variable.
2. 2. this can be used in constructor chaining. this() and super() must be the first statement.
3. 3. this can be used to invoke current class method implicitly.
4. 4. this can be used to return current instance of the class.
5. 5. this can be used as a parameter in constructor call.
6. 6. this can be used as a parameter in method call.

String is immutable in java

String is sequence of characters.String is an object in java. String objects are immutable.

Once string object is created in constant pool. They cant modified. In any change existing object will be result into a new object.

How to create String object?

1.using String literals

String str1=”helloworld”;

String str 2 =”helloworld”;

One string object and one reference variable created in constant pool. One object only place in constant pool.

**String literal/constant pool:** is a special part of heap memory used to store [string](http://tutorialspointexamples.com/string-handling-in-java/) literals or [string](http://tutorialspointexamples.com/string-handling-in-java/) constants.  
Every time a literal is created, JVM checks the [string](http://tutorialspointexamples.com/string-handling-in-java/) constant pool for it. If [string](http://tutorialspointexamples.com/string-handling-in-java/) literal is already in the pool then no new [object](http://tutorialspointexamples.com/string-handling-in-java/) will be created in the pool, a reference of the already existing [object](http://tutorialspointexamples.com/string-handling-in-java/) will returns. If [string](http://tutorialspointexamples.com/string-handling-in-java/) literal is not exist in the [string](http://tutorialspointexamples.com/string-handling-in-java/) constant pool then new instance will be created and placed in the [string](http://tutorialspointexamples.com/string-handling-in-java/) constant pool. - See more at: http://tutorialspointexamples.com/string-handling-in-java/#sthash.kZ8okKZy.dpuf

2.using new keyword

String str1=”helloworld”;

String str 2 =new String(”helloworld”);

Two objects and one refrence variable is created.one obj will be in constant pool and other obj will be heap memory .

Suppose n reference variables refer to one object “javawithease” .If one reference variable changes the value of the object, it will be affected to all other n-1 reference variables. That’s why string objects are immutable in java. –

## Summer Training

Because java uses the concept of string literal. Suppose there are 5 reference variables, all refers to one object "sachin".If one reference variable changes the value of the object, it will be affected to all the reference variables. That is why string objects are immutable in java.

String Buffer:

Its is synchronized. Its is thread safe

Its is mutable sequence of characters.

object can be modified

StringBuffer not override equals method of object class, so StringBuffer objects should be converted into string objects if you want to compare StringBuffer class objects.

String builder

It is not synchronized

It is non thread safe

StringBuilder not override equals method of object class, so StringBuffer objects should be converted into string objects if you want to compare StringBuffer class objects.