Java(References :learn java on codeacademy)

Java is platform independent language in different operating system it doesn’t need any separate installation…(c language is platform dependent)

Static and object oriented programming

Static means memory allocation is same so when we declare a variable as static static int a=10; next line if we assign a=5;then the value will change to 5

In java we use main function as static to start running from command prompt …

So we should access all function and variables using object oriental concept by creating an object.

Like Sample s=new Sample();

public class Static {

     int a=10;

     static int b=200;

    public static void main(String arg[])

    {

        Static h=new Static();

        h.simpleDisplay();

        StaticDisplay();

       // System.out.println(h.a);

    }

    void simpleDisplay()

    {

        System.out.println(a);

        System.out.println(b);

    }

    static void StaticDisplay()

    {

        System.out.println(b);

    }

}

Static function can be accessed with out object

Non static methods cannot be accessed in static area

With out creating an object.

Static is pass by reference type…

Non static is pass by value…

public class StaticExample {

    // Java program to demonstrate that

// In both static and non-static methods,

// static methods are directly accessed.

  static int num=10;

  static String  head="Hello static";

    public static void main(String arg[])

    {

     StaticExample stObj=new StaticExample();

     stObj.nonStaticdisplay();

     display();

    }

    static void display()

    {

        System.out.println("Static number:"+num);

        System.out.println("Static string :"+head);

    }

    void nonStaticdisplay()

    {//static method can be accessed in a non static method

        display();

    }

}

Syso+CTRL+SPACE will automatically print system.out.println();

Constructors

Constructor is same as other methods in java and it has same name as class name, but Doesn’t have return type. It is created at the time when object of class is created and use to add some function at the time of start running…

Interview question

Java can be written without main program because it can be run with the help of static block it is a group of statement which get executed only when class is loaded into the memory by java class loader, it is also known as static initialization block. Static initialization block is going directly into stack memory.

Inheritance

Single -only one class inherited

Multilevel

Hierarchial

* Why multiple inheritance is not in java?

Multiple inheritance means deriving A and B class to one C class is multiple inhetritance

Compiler and developers getting confused on different methods inherited by same name from different classes....

* If a constructor is defined in a class
* Class A is base class and class B is subclass

Class A class B extends class A

{ {

A() b()

{ {

System.out.println(“Hello”); Sys.O.pln(“hei”);

}

} }

}

Class c

{

public static void main(String arg[])

{

B objB=new B();

}

}

In class c object of B created constructor of that particular class get invoked but here Class B has inherited Class A so first class A constructor get invoked first and then Class B constructed get invoked..

Polymorphism

Using same function name with different parameters is **method overloading..**

**Void display**()

{System.out.println(“Hello”);}

**Void display**(int num)

{

if(num>0)

{ System.out.println(“hey it is ok”); }

}

**Method overriding**

In Class A (Base class) a method is written as **display**() then another class is Class B(Sub class) another method is written **display**()

If class B extends class A then function in class A is made as hidden and function in class B will be used…

This is called method overriding.

**Super**

Super is another key word to access base class function and variables.

If we are accessing function and variables in subclass, functions in

base class is made hidden by method overriding .And again if we are in need to use the function in base class or else the variables of base class we use a keyword **Super** to access the base class variables…

**Super.Display();**

**Super.num;//**This is variables from base class

* **Constructor with parameter**

In a classA we are using constructor with parameter and without parameters. Another class B is also having constructor with parameter and without parameters…

* Class B has inherited Class A
* Create one object of class B, so that constructor get invoked
  + If we are creating an object without argument then, constructor of class A without arguments gets invoked first then constructor of class B gets invoked.
  + If we are creating an object of class B with argument

Then constructor of class A without argument gets invoked first then constructor of class B with argument gets invoked. (it will not invoke class B constructor without argument and class A constructor with arguments)

* If we need to make call of constructor with argument of class A

Then should write

ClassB(int value)

    {

        super(10);

    System.out.println("Argument constructor of B");

    }

super(10); in constructor with argument of class B..

super should be its first statement….

**String**

**String** name=”Remy.M.Ali”;

No size definition is needed in java…

**Abstract class**

It can write function and abstract function

Interface class

An interface is reference type in java. It is similar to class .

It is a collection of abstract methods..

only declaration of function in interface class.

It is implemented with the help of implement key word.

**Package**

**Java Packages**

* Java Packages & API. A package in Java is used to group related classes. ...
* Built-in Packages. The Java API is a library of prewritten classes, that are free to use, included in the Java Development Environment.
* Import a Class. ...
* Import a Package. ...
* User-defined Packages. ...n

In package when we type Scanner and an **import** code get added to header

import java.util.Scanner;

Scanner sc= new Scanner(System.in);

**Access specifier**

Public -class methods accessible outside all packages…

Private-class can be accessed inside that class

Protected-In the same class and all inherited classes

Default-In the same class and inherited class and also

in packages..

**Exception try and catch**

Try ,catch ,throw, throws, finally..

public class tryCatchException {

    public static void main(String ar[])

    {

    try {

        int [] marks={12,23,45};

        System.out.println(marks[10]);

    } catch (Exception e) {

        System.out.println("Something went wrong");

    }

    }

}

**Thread**

* Extend thread class
* Implement Runnable interface

S.start();

Causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.

The result is that two threads are running concurrently: the current thread (which returns from the call to the start method) and the other thread (which executes its run method).

It is never legal to start a thread more than once. In particular, a thread may not be restarted once it has completed execution.

* **Throws:**
  + IllegalThreadStateException - if the thread was already started.
* **See Also:**

Thread in java pls refer class Hello and SampleThread in crossroad100k folder..