**Notes: First you prepare how to write a code for syntax and write a code**

**Java: (day one)**

install java

install eclipse

create workspace

create project

File -> Project

We call Project is program

create .java file/class

Project -> new class and give extension .java

Class Employee {

}

**How to create packages and what is best way to give name?**

From solution explorer, select project, right click and select package

Ex: companyname.projectname.foldername (this is common naming standard)

**Q)what is main method will do?**

Main method is starting point of program

**Q)what is data type and different data types**

It will represents what type of

Int

Double

Float

**Q)creating property/data members**

int salary

**Q)creating method with void**

Public static int methodname()

Public static - access modifier

int - return type

**Q)creating method with void and parameter**

Public static int methodname(int a, int b)

int a, int b  - list of parameters

**Q)creating method with return data type**

public int getArea()

{

return width\*height;

}

creating method with return data type and parameter

**Q)creating variable**

import  java.util.scanner;

class Program1

{

public  static void main(String args[])

{

system.out.println(“Hello World”);

}

}

**Q)creating static property**

Public class program2

{

Public static string name=”I am static variable”;

}

**Q) creating static method?**

 Public class StaticDemo

{

  Public static void main(String args[])

   {

    Abc.show();//calling static method

     }

}

Class Abc

{

  Public static void show() //static method

   {

 System.out.println(“Hi”)

    }

}

https://www.youtube.com/watch?v=-TxHjyC8WY4

**Q) creating object?**

 Classname obj=new Classname();

Ex: Animal b1=new Animal();

**Q) calling method with no return**

**Q) calling method with no return and parameter**

**Q)calling method with return and no parameter**

**Q)calling method with return and parameter**

**Q)calling method with return and storing the return data**

**Q)calling static method**

Public class StaticDemo

{

  Public static void main(String args[])

   {

    Abc.show();//calling static method

     }

}

Class Abc

{

  Public static void show() //static method

   {

 System.out.println(“Hi”)

    }

}

**Q)using static property**

**Java: Day2**

**Q)create classes under multiple packages**

**Q)calling classes under different packages**

**Q)write code to handle exceptions with try/catch/finally**

**A.Code for try and Catch**

**import java.util.\*;**

**import java.io.\*;**

**public class Excep\_ex5**

**{**

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

**static PrintStream out=new PrintStream(System.out);**

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

**{**

**int x, y, res;**

**try**

**{**

**out.println("enter the x");**

**x=in.nextInt();**

**out.println("enter the y");**

**y=in.nextInt();**

**res=x/y;**

**out.println("Quotient    :" + res);**

**}**

**catch(Exception e)**

**{**

**if( e instanceof ArithmeticException)**

**out.println("Exception    :" + e.toString() );**

**else if( e instanceof InputMismatchException )**

**out.println("Exception    :" + e.toString() );**

**}**

**}**

**}**

**Code for Catch and Finally keyword:**

**package javaBasics;**

**import java.io.PrintStream;**

**public class Excep\_4**

**{**

**static PrintStream out=new PrintStream(System.out);**

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

**{**

**int a, b, res=0;**

**a=10;**

**b=2;**

**try {**

**res=a/b;**

**}catch(Exception e) {**

**System.err.println(e.getMessage() );**

**}**

**finally**

**{**

**out.println("From Finally Block");**

**out.println("Result    :"+res);**

**}**

**}**

**}**

**Q)what is final keyword**

Final is reserved keyword in java to restrict the user and it can be applied to member variables, methods, class and local variables.

If you make a variable as final you can not change the value of final variable.

http://www.javatpoint.com/final-keyword

**Q)write code for interface and create class to implement that interface**

**Q)write code for creating abstract class**

To create an abstract class, you use the abstract on the class declaration and include at least one abstract method.

Ex: public abstract class Ball

     {

      Public abstract int hit(int batspeed);

     }

http://www.dummies.com/programming/java/create-an-abstract-class-in-java/  
**Q)implement method overloading**

**A.Code for method overloading:**

**package javaBasics;**

**/// example on method over-loading**

**import static java.lang.System.out;**

**class Box**

**{**

**double l, b, h;**

**public void setData(double x, double y, double z)  {**

**l=x;**

**b=y;**

**h=z;**

**}**

**public void setData(double x)  {**

**l=b=h=x;**

**}**

**public void setData()  {**

**l=b=h=1;**

**}**

**public double volume()  {**

**return(l\*b\*h);**

**}**

**}**

**public class Testclass\_9**

**{**

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

**{**

**Box b=new Box();**

**b.setData(1,2,3);**

**out.println("Volume :"+ b.volume());**

**Box bb=new Box();**

**bb.setData(3);**

**out.println("Volume :"+ bb.volume());**

**}**

**}**

**Q)implement method overriding**

**Q)implementing polymorphism**

**Q)implementing interface**

**Q)write a code to save data into excel file and read from excel file (POI and jexcel API)**

**Q)how to update the data into XML file and read data from XML file**

**Java: Day3**

**Q)write code to add items to integer, string array**

**Q)write code to retrieve items from integer, string array**

**Q)write code to add items to ArrayList collection**

**Q)write code to retrieve items from arraylist**

**Q)write code to add items HashMap**

**Q)write code to retrieve items HashMap**

**Q)write code to connect to JDBC to get rows from employee table**

**Q)create Employee class**

**Q)Add employee class to list collection**

**Q)create method that return list of employee collection**