1. what is main method will do?

**Ans:** Main method is starting point of program

1. What is variable?

**Ans:** It will store the value in memory

To create variable, we specify

Datatype variableName;

1. what is data type and different data types?`

**Ans:** It represents --- which type of values can be stored

--- What will be the size of the value, how much memory will be allocated

There are 8 types of data types

Byte, Short, INT, Long, float, double, char, Boolean.

1. Creating method with void:

**Ans:** we write methods in void

public void method1() {

}

1. Creating method with void and parameter?

**Ans:**

public void method1(int) {

}

1. Creating method with return data type?

**Ans:**

public int getArea () {

return width \* height;

}

1. creating method with return data type and parameter?

**Ans:**

public int getArea(int) {

return width \* height;

}

1. creating variable?

Int A;

1. creating static property: All instances shared the value

http://crunchify.com/java-static-methods-variables-static-block-and-class-with-example/

1. creating static method?

Public static method1() {

}

1. creating object?

Bharath b = new Bharath ();

1. calling method with no return?

b.getArea();

1. calling method with no return and parameter?

b.getArea(int);

1. calling method with return and no parameter?
2. calling method with return and parameter
3. calling method with return and storing the return data
4. calling static method?

static fields and methods can be called by using the class name

1. using static property:

**Ans:** public static int empid;

**Java: Day2**

create classes under multiple packages

calling classes under different packages

write code to handle exceptions with try/catch/finally

what is final keyword

write code for interface and create class to implement that interface

write code for creating abstract class

implement method overloading

implement method overriding

implementing polymorphism

implementing interface

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

how to update the data into XML file and read data from XML file

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

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

write code to add items to ArrayList collection

write code to retrieve items from arraylist (using for each loop\_

write code to add items HashMap

write code to retrieve items HashMap

Write code to add items to hashset

Write code to retrieve items to hasset

write code to connect to JDBC to get rows from employee table

Write method to return list of rows code to loop throughs

Q. create Employee class?

Class Employee {

}

Add employee class to list collection

create method that return list of employee collection

Difference between string, string buffer, string builder with example?

String  
  
String is immutable (once created cannot be changed) object. The object created as a String is stored in the Constant String Poo.    
Every immutable object in Java is thread safe, that implies String is also thread safe. Two threads cannot use string simultaneously.  
String once assigned cannot be changed.

String demo = " hello ";

StringBuffer  
  
StringBuffer is mutable means one can change the value of the object. The object created through StringBuffer is stored in the heap. StringBuffer has the same methods as the StringBuilder, but each method in StringBuffer is synchronized that is StringBuffer is thread safe.

StringBuffer demo1 = new StringBuffer("Hello");  
// The above object stored in heap and its value can be changed.

demo1=new StringBuffer("Bye");

StringBuilder  
  
StringBuilder is same as the StringBuffer, that is it stores the object in heap and it can also be modified. The main difference between the StringBuffer and StringBuilder is that StringBuilder is also not thread safe.   
StringBuilder is fast as it is not thread safe.

StringBuilder demo2= new StringBuilder("Hello");  
// The above object too is stored in the heap and its value can be modified  
demo2=new StringBuilder("Bye");