1. What will main method do...?

The program execution starts from main method. A program can be compiled without main method but for program to run then main method is must.

2.Create property/data members..?

Properties can be created using different datatypes depending upon the class they are.

3. What is datatype...?

Datatype defines the type of variable

different types of datatypes are int, char, float, double,long,String and etc..

4. what is variable..?

Variable is a piece of memory that has a data value.

5. Create method with void..?

Method with a return type void does not return any value.

6. Creating method with return type?

We can return any data type like int, string , float boolean and etc...

7. Create default/parameterized constructors..?

The constructors are given class name they donot have any return type. Constructors can have no parameters or have parameters..

public classA(){

}

8.Create method with return datatype and parameter...?

public int add(int a, int b){

int sum;

sum = a+b;

return sum

}

9 Static property :

static property is meant for the class. it cannot be instantiated by any instance of the class

10. Static block:

Static block is executed even before the main method

Static {

}

11. calling method with void

public ClassA{

int num;

public void setnum(int a){

}

public static void main(){

ClassA c1 = new ClassA(); #Creating object

c1.setnum(5);

}

}

12.Calling method with return and no parametr

public ClassA{

int num;

public void setnum(int a){

}

public int getnum(){

return num;

}

public static void main(){

ClassA c1 = new ClassA(); #Creating object

c1.setnum(5);

sysout(c1.getnum());

}

}

13. calling static method

Only a static methd can call a static method

14. Creating classes under multiple packages?

package A

public Class A{

}

package B

public Class B{

}

15. Calling classes under different packages

we can call classes from different packages by importing the package into the current class.

16. Code to handle exceptions using try /catch??

public Class A

public static void main(String []args){

a =100;

b=0;

try{

quot?ient =a/b;

sysout(quotient);

catch(Exception e){

e.printstacktrace(){

}

}

17. Checked Exceptions and unchecked exceptions?

\*Checked exceptions are compiletime exceptions

\*Unchecked exceptions are runtime exceptions.(logical errors)

18.Final Key word?

Final keyword is used when the data is not changed. once a variable is declared final it cannot be changed.

A final method cannot be over ridden.

Public final Class Employee{

public final int empid;

public final void getdetails(){

}

}

19.Abstract class

public abstract ClassA{

}

20.implement method overloading?

changing method parameters.

Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different.

}class DisplayOverloading

{

public void disp(char c)

{

System.*out*.println(c);

}

public void disp(char c, int num)

{

System.*out*.println(c + " "+num);

}

}

class Sample

{

public static void main(String args[])

{

DisplayOverloading obj = new DisplayOverloading();

obj.disp('a');

obj.disp('a',10);

}

}

21. implement method over riding?

Declaring a method in **subclass** which is already present in **parent class** is known as method overriding. Changing the method body.

My baseclass {

Public void disply();

{

System.out.println(“parent class method”);

}

}

Class Mychildclass extends mybaseclass{

Public void display(){

System.out.println(“child class method”);

}

Public static void main ( String args[] ){

Mychildclass obj =new Mychildclass();

Obj.disp();

}

22.implementing polymorphism?

Both method ovrloading and overriding together implements polymorphism

23.implementing interface?

Interface consists of all abstract methods.

public Interface A{}

24. Inheritance in Java

**Inheritance**is one of the feature of Object-Oriented Programming ([**OOPs**](http://beginnersbook.com/2013/04/oops-concepts/)). Inheritance allows a class to use the properties and methods of another class. In other words, the derived class inherits the states and behaviors from the base class. The derived class is also called subclass and the base class is also known as super-class. The derived class can add its own additional variables and methods. These additional variable and methods differentiates the derived class from the base class.

Inheritance is a [**compile-time**](http://beginnersbook.com/2013/04/runtime-compile-time-polymorphism/) mechanism. A super-class can have any number of subclasses. But a subclass can have only one superclass. This is because Java does not support multiple inheritance.

Use extends keyword

Public Class A Extends B{}

25.write code for retrieving integer and string array iems?

public Class A{

int StringArray[] = {"red","white", "blue"};

public static void main(){

for(i=0; i<StringArray.length; i++){

sysout(i);

}

}

26.Write Code to add Items to ArrayList Collection?

public class A{

public static void main(String[] args){

ArrayList<String> al = new Arraylist();

al.add("red");

}

}

27.retrieve items from ArrayList

public class A{

public static void main(String[] args){

ArrayList<String> al = new Arraylist();

al.add("red");

al.add("blue");

al.add("white");

for(i=0;i<=al.size();i++){

sysout(i);

}

}

}

28.Code to add items to HashMap and retrieve items.

public class A{

public static void main(String[] args){

Map<integer,String> Map = new HashMap();

map.put(100,"Vennela");

map.put(200,"Sneha");

map.put(300,"Bindu");

Set<integer> s1 = map.keyset();

Iterator it = it.next();

int temp = it.next();

sysout(temp);

sysout(map.get(temp));

}

29.Code to implement hashset

public class A{

public static void main(String[] args){

Set<String> s1 = new HashSet();

s1.add("shahsi");

s1.add("Sneha");

s1.add("Bindu");

Iterator<String> it = s1.iterator();

sysout(it.next());

sysout(it.next());

sysout(it.next());

}

}

30.JDBC Connection

Public Class Employee{

int empid;

String empname;

public Static void main()

Connection conn;

conn = DriverManager.getConnection("jbbc:oracle:thin@localhost:1521:xe", testavco;xxxx24);

}

31. Create Employee Class

public class Employee{

int empid;

int empname;

public static void main(String[] args){

Employee e1 = new Employee();

Employee e2 = new Employee();

List<Object> l1 = new ArrayList();

l1.add(e1);

l1.add(e1);

for(i=0; i<;1.size(); i++){

student x = li.get(i);

}

}

}

32.String Buffer and String Builder

String: Character Strings cannot be modified where as string buffer represent character strings that can be modified.

String buffer is synchronized where as string builder is not..