COREJAVA Interview Questions

Q1. Can we define static variables inside the static method or not?

Ans: No, not possible, because once we defined the varaibles in methods will become local type.

```
static void add()
{       static int  num = 100;
}
```

Q2. In project when we will instance & static varaibles give the scenario?

Ans: instance variables every object separate memory created.
where as static varaibles for all object single copy created per class.

```
class Emp
{       int eid;
        String ename;
        double esal;

        public static final  String company="tcs";
        public static final  String location="Hyderabd";
}
```

Q3. when we will use for vs. for-each?

Ans:
for loop it is possible to apply some conditions. start end inc/dec

```
int[] a = {1,2,3,4....1000};
for (int i=333;i<=444;i+=5)
{       System.out.println(a[i]);
}
```

for-each loop conditions are not apply it will print the data from start to end.

```
int[] a = {1,2,3,4....1000};
for (int x : a)
{       System.out.println();
}
```

Q4. Define JDK vs. JRE vs. JVM?

Ans:

JDK will provides both compile time & runtime environment, in Development we will use JDK.

JRE will provides only runtime environment, in production we will JRE.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

JRE internally contains JVM to give the the execution environmet.

Q5. How many parts of java?

Ans: There are three parts,

J2SE : java 2 standard edition : Standalone apps
J2EE : java 2 enterprise edition : web,enterprize apps
J2ME : java 2 micro edition :mobile apps

Q6. What is the purpose of methods?

How many types of methods?
How to access the methods in java?

Ans :
Methods are used to write the business logics of the application.

There are two types of methods in java,

a. instance method

```
void add(){
}
```

b. static method

```
static void add(){
}
```

Access the instance method using object-name.
Access the static method using class-name.

Instance method are possible to override.
static methods not possible to override.

Q7. Define class vs. Object.

Ans: class is a logical entity contains the logics of the application.
object is a physical entity represents the memory.

class is a blueprint it decides the object creation.Without class unable to create the object.

object is real world entity contains properties & behaviours is called object.
single class possible to create multiple objects, every object needs memory.

Declare the class using class keyword.
Create the object using new keyword.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q8. Define and(&) , or(|)?

Ans: and : All conditions are true then it returns true.

```
if (username.equals("ratan") & password.equals("anu"))
{       body
}
```

or : Any one is true then it returns true

```
if (company.equals("tcs") | salary>=100000)
{       body
}
```

Q9. Define the constructor?

How many types of constructors in java?
What is the advanatages of constructor?

Ans: Constructor is a special method to write the logics, these logics are automatically executed during object creation.
constructor name & class-name should same.
No return type but arguments are allowed.

There are two types of constructors:

a. default constructor : compiler generated : 0-arg cons with empty impl
b. userdefined constructor
i. 0-arg cons
ii. prams cons

Constructors are used to initialize the data durng object creation.

```
class Emp
{       int eid;
        String ename;
        double esal;
        Emp(int eid,String ename,double esal)
        {       this.eid = eid;
                this.ename = ename;
                this.esal = esal;
        }
        public static void main(String[] args)
        {
                Emp e = new Emp(111,"ratan",70000.45);
        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q10. define break vs. continue?

Ans: break : Break is used to stop the loop execution.

```
for (int i=1;i<=10 ;i++ )
{       if (i==5)
        {       break;
        }
        System.out.println(i);
}
output:  1 2 3 4
```

we can use break statement in two places,

a. inside the switch.

b. inside the loops.

continue : used to skip the particular iteration.

```
for (int i=1;i<=10 ;i++)
{       if (i==5)
        {       continue;
        }
        System.out.println(i);
}
output: 1 2 3 4 6 7 8 9 10
```

Q11. Define the package in java?

What are the elements present in package?

what is the default package in java?

Ans: java predefined support in the form of packages,

it is a collection of elements.

The package contains 6-elements,

classes,interfaces,enums,annotations,exceptions,error

To check the API docs use below link,

https://docs.oracle.com/javase/8/docs/api/

The java contains 14-packages, the default package in java is : java.lang

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q12. What is the difference between bitwise & logical operator?

Ans:

Bitwise operators : & |
Logical operator : && ||

Bitwise operators can be used for bitwise calculations & logical conditions

```
sop(10&5)                    // 6          // int
SOP(10>20 & 30<40)  // false      // boolean
```

Logical operators can be used for only Logical conditions check.

```
SOP(10&&5)                   // Invalid
SOP(10>20 && 30<40)  // false      // boolean
```

Bitwise operator check the both condition then return result.
Logical operators second condition execution depends on first condition.

Q13. What are the entry controlled loop & exit controll loop?

Ans: *Entry controlled loop : for,while*

```
for (initialization;condition ;inc/dec)
{       body
}

while (condition)
{       body
}
```

Exit controlled loop : do-while

```
do
{       body
}while (condition);
```

in for,while loop the mminimum execution: 0-times
in do-while the minimum execution : 1-time

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q14. When we will get StackOverflowError?

Ans: *The stack memory will be created by JVM before calling main method.*
The stack memory will be destroyed by JVM after completion of main method.

```
void main()
{       add(10,20);
        mul(100,200)
}
```

when we call the methods cyclic.
m1 is calling m2 :: m2 is calling m1 keep on storing in stack we will get StackOverflowError.

```
void m2()
{       m1();
}
void m1()
{       m2();
}
void main()
{       m1();
}
```

Q15. Explain instance & static blocks? What is the purpose of blocks?

Ans:

a. static blocks are executed only once when class file is loading.
b. instance blocks are executed during object creation before constructor execution.
constructor logics are specific object but instance blocks logics are comman to all objects.
d. Inside the class possible to declare the more than one instance & static blocks.
The execution order is top to bottom.

```
class Test
{       static
        {       System.out.println("static block");
        }
        {       System.out.println("instance block...");
        }
        Test()
        {       System.out.println("0-arg constructor...");
        }
        Test(int a)
        {       System.out.println("1-arg constructor...");
        }
        public static void main(String[] args)
        {       Test t1 = new Test();
                Test t2 = new Test(10);
        }
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

```
}
```

Q16. What are the different ways to load the .class file into memory?

Ans: a. when we create the object the class loaded then the static block executed.

b. Wihtout object creation just to load the class programatically use forName() method, it is a static method of Class.

c. forName() method throws ClassNotFoundException it is a checked exception so must handle the checked exception using try-catch or throws keyword.

```
class Demo
{       static
        {       System.out.println("Demo class static block");
        }
}
class Test
{       public static void main(String[] args)throws Exception
        {       //Demo d = new Demo();
                Class.forName("Demo");
        }
}
```

Q17. How to call the constructor in java?Is it one constructor can call multiple constructors?

Ans: To call the constructor use this keyword.

```
this(10)                    : calling 1-arg constructor
this(10.5,'a')              : calling 2-arg constructor
this("ratan",false,20.3f)   : calling 3-arg constructor
```

Inside the constructor, the constructor calling this keyword should be first line. So one constructor can call only one constructor.

```
class Test
{       Test()
        {       this(10);
                System.out.println("0-arg cons");
        }
        Test(int a)
        {       this(10,20);
                System.out.println("1-arg cons");
        }
        public static void main(String[] args)
        {       new Test();
        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q18. What is method recursion?

Ans: Method recursion : method calling itself is called recursion?

```
class Test
{       static void validate(int num)
        {       if (num>0)
                {       System.out.println("My Number is..."+num);
                        validate(--num);
                }
                else
                {       System.out.println("Your number is Negative....");
                }
        }
        public static void main(String[] args)
        {       Test.validate(10);
        }
}
```

Q19. What are the Different ways to call the static members in java?

Ans: we can access the static variables in three ways,

a. Using class-name [it is recommanded]
b. Direct access
c. Access using object name.

```
class Test
{       static int num = 10;
        public static void main(String[] args)
        {       System.out.println(Test.num);

                System.out.println(num);

                Test t = new Test();
                System.out.println(t.num);
        }
}
```

Note: Local variable access directly.
Static variables access using class-name.
Instance variables access using object-name.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q20. When we will use switch & else-if statements in Application?

Ans: when we want check multiple conditions then we have to use else-if.

```
if (cond1){
}
else if (cond2){
}
else if (cond3){
}
else{
}
```

in switch based on the argument data the matched case will be executed.

```
switch(argument)
{   case label-1     :     statement(s); break;
        case label-2 :     statements;   break;
        default      :     statements;  break;
}
```

Q21. When we will use for vs. while loops in Application?

Ans: case 1:

for loop is recommended when we have the : start , end_cond , incre/decre

```
for (initialization;condition;inc/dec)
{       Logics Here....
}
for (int j=1;j<=10 ;j++)
{       System.out.println("Good Morning....");
}
```

while loop is recommended when we have the only condition check.

```
while (condition){
}
ArrayList names = {"ratan","anu","sravya"};
while(names.hasNext())
{       System.out.println(names.next());
}
```

case 2:

if you know number of iterations use for loop. :: for(int i=1;i<=5;i++)

if you do not no number of iterations use while loop :: while(true)

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q22. Can we access the other packages classes without import statements?

Ans: It is possible to access other package classes without import using full-name of the class.

```
package com.tcs;
public class A{
}
//application with import :  Access the class directly
package com.wipro;
import com.tcs.A;
public class Test
{       A a = new A();
}
//application without import : Access the class using full-name.
package com.wipro;
class Test
{       com.tcs.A a = new com.tcs.A();
}
```

Q23. What is the difference between the normal import & static import?

Ans: There are two types of imports

a. normal import

Here we can access both instance & static data. But here access the static data using class-name.

```
import com.tcs.info.Test;
```

b. static import

using static import it is possible to access only static data directly without using class-name. It is not possible to access instance data.

```
import static com.tcs.info.Test.*;
```

Q25. What are the modifiers applicable to constructors?

Ans: In java to the constructors only four modifiers are applicable,

public
private
protected
default(by default)

If the class contains private constructor, it is not possible to create the object outside of the class.

We can prevent the object creation of outside the class by declaring the private constructor.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q25. Explain System.out.println()?

Ans: System.out.println();

System ---> It is a class present in java.lang

out ---> it is static variable of System class

public static final java.io.PrintStream out;

println() ---> It is method present in PrintStream to print the data.

case 1:

```
class Test
{       Employee emp;
}
```

Note: emp is a variable of Test class of type Employee.

case 2:

```
class System
{       public static final java.io.PrintStream out;
}
```

Note: out is a variable of System class, of type PrintStream.

Q26. Can we access sub-packages data when we import main package with *?

Ans: No, not possible.

The main packages contains sub packages also.

java.lang
java.lang.annotation
java.lang.instrument
java.lang.invoke
java.lang.management
java.lang.ref
java.lang.reflect

```
import java.lang.*;
```

*When we import main package with * it is possible to access only main package classes,*

but not sub packages classes, to Access the sub package classes import sub packages also.

```
import java.lang.*;
import java.lang.annotation.*
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q27. What are the Permission/Scoping modifiers in java?

Ans: There are four Permission/Scoping modifiers in java,

public : classes methods variables constructors : All packages can access.

default: classes methods variables constructors : Only with in the package can access.

private : methods variables constructors : only with in the class.

protected : methods variables constructors : default + outside package only child classes.

Permission from high level to low level:

public ---> protected ---> default ---> private

The most accessible modifier in java : public

The most Restricted modifier in java : private

The default modifier in java : default

Q28. Define inheritance? how many types inheritance in java?

Ans:

The process of creating new class by taking the properties from existing class is called inheritance.

We are achieving inheritance concept by using extends keyword.

Inheritance is also known as "is-a" relationship.

There are 5-types of inheritance in python,

i. Single inheritance : one parent with one child.

ii. Multi level inheritance : generation by generation.

iii. Hierarchical inheritance: one parent with multiple child classes.

iv. Multiple inheritance : multiple parent classes with one child.

v. Hybrid inheritance : hierarchical + multiple

In general we have five types of inheritance but java support three types.

In java one class can extends only one class at a time. If we are extending more than one class we will get error message.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q29. Explain about Object class in java?

Ans: *important points about Object class.*

```
class A{
}
class B extends A{
}
class C extends B{
}
```

a. The default super class in java is : Object
b. The root class in java is : Object
c. Every class contains parent class except : Object
d. Every class in java is Child of Object either directly(A) or indirectly(B,C).
e. Object class present in "java.lang" package.This class contains 11 methods ,so all classes can use that 11-methods because Object is root class.

Q30. Define the Polymorphism? How many types of Polymorphisms in java?

Ans: *One functionality with different behaviors is called polymorphism.*
The ability to appear in many forms is called polymorphism.
Poly(many) + Moprphsim(forms) = many forms.

case-1: sleep() is one method having two different bahaviours.
sleep(1000)
sleep(1000,2000)
case 2: wait() is one method having three different bahaviours.
wait()
wait(1000)
wait(1000,2000)

There are two types of polymorphism in java,

1) *Compile time polymorphism / static binding / early binding*
   ex : method overloading.
2) *Runtime polymorphism /dynamic binding / late binding.*
   ex : method overriding.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q31. What are the Different types of overloading in java? What is the advantage?

Ans: *There are three types of overloading in java,*

a. Method overloading.
b. Constructor overloading.
c. Operator overloading.

Method overloading:

```
class Test
{       //overloaded methods  :  same method name but different number of arguments.
        void m1(int a){}
        void m1(int a,int b){}
      //overloaded methods : same method name & same number of arguments but different types.
        void m2(String name){}
        void m2(char ch){}
}
```

Constructor overloading:

```
class Test
{       Test(int i)      {System.out.println("int argument constructor");}
        Test(char ch,int i)     {System.out.println("char,int argument constructor");}
}
```

Operator Overloading:

```
System.out.println(10+20);                  //30          [addition]
System.out.println("ratan"+"anu");          //ratananu    [concatenation]
```

```
//Applicaiton without overloading : number of method names more
class Test
{       void add2(int num1,int num2){     }
        void add3(int num1,int num2,int num3){ }
        void add4(int num1,int num2,int num3,int num4){       }
}
//Application with overloading : we can use same name with different behaviour.
class Test
{       void add(int num1,int num2){       }
        void add(int num1,int num2,int num3){   }
        void add(int num1,int num2,int num3,int num4){        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q32. What is method Overriding? what is the advantage?

Ans: *Re-writing the parent method implementation in child class according to the child specific implementation.*

Parent method is called overridden method.

Child method is called overriding method.

To override the methods we need two classes with parent & child relation.

```
class Parent
{       void eat()
        {       System.out.println("idly.....");
        }
}
class Child extends Parent
{       void eat()
        {       System.out.println("poori/ dosa.....");
        }
        public static void main(String[] args)
        {       Child c = new Child();
                c.eat();
        }
}
```

Q33. What are the rules to fallow while overriding the method?

Ans: *Overriding Rules: 9-rules*

1. *Overridden method signature & overriding method signature must be same.*
2. *final methods are not possible to override.*
3. *Method return types must be same at primitive level.*
4. *Method return types can be changed at object level using Co-variant return type.*
5. *private methods are not possible to override.*
6. *Same & increasing level permissions are valid but not possible to decrease the permission.*
7. *static methods are not possible to override.*
8. *overridden method not throws any exception,*
   then overriding method can throws unchecked but not checked.
9. *overridden method throws exception then*
   overriding method can throws same exception
   overriding method not throws any exception
   overriding method can throws child exception
   overriding method can not throws parent exception

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q34. What is Co-variant return type in java?

Ans: co-variant return type :

The overriding method return type is sub type of overridden method return type.

```
class Dog {}
class Puppy extends Dog{}
class Parent
{       Dog info()
        {       return new Dog();
        }
}
class Child extends Parent
{       Puppy info()
        {       return new Puppy();
        }
}
```

Q35. What is Method overriding & method hiding in java?

Ans:

Method overriding:

instance methods we will override.
Method signature must be same in parent & child classes.
method execution decided by object.

Method hiding:

static methods we will hide.
Method signature must be same in parent & child classes.
Here method execution decided by class-name.

Observation :

a. The parent & child classes methods are instance is called method overriding
b. The parent & child classes methods are static is called method hiding.
c. In parent & child one method is instance, one method is static is not allowed.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q36. What is the advantage of parent class reference hilding the Child class object?
or
What is runtime polymorphism?

Ans: The parent class can hold child object. But child class unable to hold parent object.

When we take the method argument is parent class, then it will hold all child objects.

At runtime which object is passing that object class methods are executed. So the method execution decided at runtime is called Runtime polymorphism.

```
interface Birds
{       void fly();
}
class Parrot implements Birds
{       public void fly(){      System.out.println("parrot can fly");       }
}
class Penguin implements Birds
{public void fly()      {       System.out.println("Penguin can't fly");   }
}
class TestClient
{void info(Birds b)             //Birds b = new Parrot() ::  Birds b = new Penguin()
        {       b.fly();
        }
        public static void main(String[] args)
        {       TestClient3 t = new TestClient3();
                t.info(new Parrot());
                t.info(new Penguin());
        }
}
```

Q37. Explain about final modifier/prevention modifier in java?

Ans: *final variables are fixed constants modifications are not allowed.It is not possible to*

reinitialize the final varaibles in java.(Preventing varaibles re-initialization)

```
class Test
{       final int num = 100;
}
```

Final methods not possible to override. (Preventing Method overriding)

```
class Parent
{       final void eat(){  System.out.println("Idly");}
}
```

final classes not possible to inherit. (Preventing Inheritance)

```
final class Services
{
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q38. Define abstract methods & normal methods in java?

Ans: There are two types of methods

a. Normal methods.

b. Abstract methods.

Normal methods : Contains method declaration & implementation.

```
void add()
{   logics
}
```

abstract methods:

The abstract method contains only method declaration but not implementation.

The abstract method must ends with semicolon.

To represent the method is abstract using abstract modifier.

ex: abstract void add(int num1,int num2);

Q39. Explain normal classes & abstract classes in java?

Ans: The classes are divided into two types,

1) Normal classes.

2) Abstract classes.

Normal classes: contains only normal methods.

```
class Test
{   void add(){}
    void mul(){}
    void div(){}
}
```

Abstract class:

case 1: The class contains at least one abstract method is called abstract class.

```
abstract class Test
{   abstract void add();
    abstract void mul();
    abstract void div();
}
```

case 2: The abstract class contians zero abstract methods.

```
abstract class Test
{   void add()   {   logics...}
    void mul()   {   logics...}
    void div()   {   logics...}
}
```

The abstract class may contains abstract methods or may not contains abstract methods but for the abstract classes object creation is not allowed.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q40. what is the abstraction concept in java? What is the advantage of abstraction?

Ans:

Abstraction is a process provides service details but not its implementation details.

Data abstraction is the process of hiding certain details and showing only essential information to the user.

Data abstraction is the process of hiding implementation details and showing service details.

Abstraction can be achieved with either abstract classes or interfaces.

Q41. Explain about interfaces in java?

Ans:Interfaces are used to declare the functionalities of the project.

Interface provides service details but not its implementation details.

Declare the interface using interface keyword.

Interfaces also .class files are generated.

```
interfaces are by default                 :   abstract
interface methods are by default  :   public abstract
interface variable are by default  :   public static final
    interface Bank
    {   int limit = 40000;
        void roi();
    }
    class AxisBank implements Bank
    {   public void roi()
        {   System.out.println("Axis Bank ROI 9.99.... withdraw Limit"+Bank.limit);
        }
    }
    class TestClient
    {   public static void main(String[] args)
        {   AxisBank ax = new AxisBank();
            ax.roi();
        }
    }
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q42. What is the difference between interfaces & abstract classes & normal classes & client code?

Ans: In real time the Project structure shown below.

Level 1 - interfaces : contains the service details
Level 2 - abstract classes : contains partial implementations
Level 3 - implementation class : contains all implementations
Level 4 - client code : Access the data.

```
interface Bank
{       void deposit();
        void withdraw();
}
abstract class Dev1 implements Bank
{       void deposit(){implementation here}
}
class Dev2 extends Dev1
{       void withdraw(){implementation here}
}
class TestClient
{       public static void main(String[] rgs)
        {       Dev2 d = new Dev2();
                d.deposit();  d.withdraw();
        }
}
```

Q43. How to clone the objects in java? What is the advantage?

Ans:1. The process of creating exactly duplicate object is called cloneing process.
2. To make the cloning process the class must implements Cloneable interface.
To create the cloning use clone() method & this method present in Object class.

```
class Test implements Cloneable
{       int num1=10;
        int num2=20;
        public static void main(String[] args)throws Exception
        {       Test t = new Test();
                System.out.println(t.num1+" "+t.num2);

                Test x = (Test)t.clone();
                System.out.println(x.num1+" "+x.num2);
        }
}
```

initially your class does not support cloneing process. So your class must implements Cloneable.

java.lang.Cloneable ----> provides cloning capabilities

java.io.Serializable ----> provides Serialization capabilities
java.util.RandomAccess ----> data accessing capabilities

The above process is deep cloning creating duplicate objects.
One more cloning is shallow cloneing. creating duplicate references.

```
Test t1 = new Test();
Test t2 = t1;                    // here t2 also is pointing to t1 object
```

Q44. What the new features about interfaces in java?

Ans:In Java SE 7 or earlier versions, an interface can have only two things i.e.

Constant variables(static variables)
Abstract methods.

From java-8 the interface allows three methods,

Constant variables
abstract method
default methods
static methods

Java 9 onwards, inside the interface we can include,

Constant variables
abstract methods
default methods
static methods
private methods
private static methods

Q45. Explain about functional interfaces in java?

Ans:The interface contians only one abstract method is called functional interfaces.

The functional interface can have multiple default & static & private & private static methods.

The lambda expression will provide the implementation of Functional Interface.

```
default void forEach(Consumer<super T>action)
```

The method having the argument(Consumer) it is a Functional interface, so we can pass the lambda expression logics as argument data. This process is called Functional programming.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q46. What is the meaning of Automatic garbage collection?

Ans:

In a programming language like C,CPP, allocating and deallocating memory is a manual process.

In Java, process of deallocating memory is handled automatically by the garbage collector.

Automatic garbage collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects.

a. An in use object, or a referenced object, means that some part of your program still maintains a pointer to that object.

b. An unused object,or unreferenced object, is no longer referenced by any part of your program.So the memory used by an un-referenced object can be reclaimed.

Garbage collector will collect the garbage. Garbage means an object without reference.

When the JVM Running,Then only the garbage collector will run automatically to destory the un-used objects.

Q47. What are the Different ways to call the Garbage Collector?

Ans: We can call the GC in two ways

i. System class : gc() : static method

ii. Runtime class : gc() : instance method

only Runtime class gc() method can call the garbage collector directly. The System.gc() internally calls Runtime class gc() to call the Garbage Collector.

```
class Test
{       public void finalize()
        {       System.out.println("object destroyed");
        }
        public static void main(String[] args)
        {       Test t1 = new Test();
                System.out.println(t1);
                t1 = null;

                Runtime r = Runtime.getRuntime();
                r.gc();
        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q48. What are the Different ways to make the object un-referenced?

Ans: There are 4-ways to make the object is un-referenced :(Giving the object to Garbage Collector)

1. *Assigning the null value to object.*

```
Test t = new Test();
t = null;
```

Note: Initially t-reference is pointing to object, later that reference is reassigned to null, so the Test object becomes un-referenced it will collect by Garbage Collector.

2. *Name less object*

```
new Test();
```

Note: Once the above line is completed, that object does not have any reference, So it will collect by Garbage Collector.

3. *Reassign reference-variable*

```
s1 ----> ratan
s2 ----> anu
s1 = s2
```

Note: s1=s2 Here both references are pointing to "anu" so the "ratan" object becomes un-referenced, So the un-referenced object will collected by Garbage Collector.

4. *When we create the object inside the method,once the method is completed object is eligible to Garbage Collector.*

```
void m1()
{       Test t = new Test();
        stack           heap
}
```

Q49. Explain Encapsulation mechanism in java?

Ans: Encapsulation in Java is a mechanism of wrapping the data (variables) and code (methods) together as a single unit.

Grouping mechanism is called encapsulation.

ex: class , package.

The main objective of encapsulation is data hiding to achieve security and it is possible to hide the data by using private modifier.

If a variable declared as a private it is possible to access those variables only inside the class is called data hiding.

Fully encapsulated class : The class contains private properties.

```
class Emp
{       private int eid;
        private String ename;
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q50. What are the different ways to initialize the data in java explain?

Ans: In java we can initialize the data in two ways,

1. Using constructor :

If the constructor taking five arguments mandatory we have to pass the 5-arguments then only object is created. In future if we want change the particular variable data is not possible.

```
class Emp
{       Emp(int eid,String ename,double esal)
        {       this.eid = eid;
                this.ename = ename;
                this.esal = esal;
        }
}
```

2. Using setter approach :

We can create the object, Depends on the values that we have, we can call specific setter methods remaining values are stored default values. In future if we want change the particular data call the particular setter methods.

```
class Emp
{       public void setEid(int eid)
        {       this.eid = eid;
        }
        public void setEname(String ename)
        {       this.ename = ename;
        }
        public void setEsal(double esal)
        {       this.esal = esal;
        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q51. Define type conversion? what are the different types of type conversions in java?

Ans:Type Conversion:The process of converting data one type to another type is called type casting.

There are two types of type casting

1. *Implicit typecasting /widening/up casting*
2. *Explicit type-casting (narrowing)/down casting*

Implicit-typecasting: (widening) or (up casting)

a. *This will performed when we assign lower data type value to higher data type.*

b. *Compiler is responsible to perform implicit typecasting.*

c. *It is also known as up-casting or widening.*

d. *When we perform up casting no data loss.*

Explicit type-casting: (Narrowing) or (down casting)

a. *This will performed when we assign higher data type value to lower data type.*

b. *Developer is responsible to perform explicit typecasting.*

c. *It is also known as narrowing or down casting.*

d. *When we perform down casting data will be loss.*

```
class Test
{       public static void main(String[] args)
        {       //implicit type conversion  :: upcasting :: compiler perform :: No loss of data
                double d = 10;
                System.out.println(d);
                int x = 'a';
                System.out.println(x);
                //Explicit type conversion  :: down casting :: developer :: Loss of data.
                int num = 130;
                byte b = (byte)num;
                System.out.println(b);
        }
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q52. Why method signature of main() method is always public static void main(String[] args) ?

Ans.

i. The main() method is automatically invoked by JVM. The JVM can invoke main() method if and only if the main() should have more visibility means less restrictive which is public modifier.

ii. In java the execution always starts with main() method i.e., before main() method we cannot get a chance to create object. In order to invoke method without creating object we have to declare such method as static.

iii. The main() method is automatically invoked by JVM so we are not required to return our project specific data to JVM. Hence return type is void.

iv. The JVM always create new String[]{} and passed as argument to main() method. Hence parameter type of main() method is always String[].

Q53. Though interface is a pure abstract class then why interfaces are needed?(OR) What if you had an Abstract class with only abstract methods? How would that be different from an interface? (OR)What are the difference between an abstract class and an interface?

Ans:

Abstract class:

The abstract classes may contain state (data members) and/or implementation (methods).

Can contains non final & non static variables i.e., abstract class may have state.

Only abstract methods are good enough to implement in subclass.

Does not supports multiple inheritance using abstract classes.

Can contains any number of constructors.

Interface:

Interfaces can have no state or implementation.

All variables must be static and final. Hence there is no state in interface.

The subclass of an interface must provide an implementation of all the methods of that interface.

Interfaces supports multiple inheritance.

Never contains constructors at all.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q54. Define the exception? What is the main objective of exception handling?

Ans: The dictionary meaning of the exception is abnormal termination.

exception is is a object raised during the execution of the application to disturb the normal flow of execution is called exception.

Whenever the exception raised in application,

a. Program terminated abnormally.

b. Rest of the application not executed.

we can handle the exception using try-except blocks.

once we handle the exeption

a. Program terminated normally.

b. Rest of the application executed.

The main objective of exception handling is to get the normal termination of application, to execute the remaining code.

Q55. what are the types of exceptions in java?

Ans: Types of Exceptions: As per the sun micro systems standards The Exceptions are divided into three types

1. Checked Exception

2. Unchecked Exception

3. Error

#Unchecked Exception:

a. The Unchecked Exception are caused due to end-user input problems.

b. The exceptions are not checked by compiler are called Unchecked Exception

ex : ArithmeticException, AIOBE , NPE , NFE...etc

c. These are child class of RuntimeException.

#Checked Exception:

a. The checked exceptions are caused due to developer issues.

b. The Exceptions which are checked by the compiler are called Checked Exceptions.

FileNotFoundException,SQLException,InterruptedException ……..etc

c. these are child class of Exception.

#errors:

1.The exceptions are occurred due to the fallowing reasons

a. Developer mistakes

b. End-user input mistakes.

But errors are caused due to lack of system resources.

StackOverFlowError, OutOfMemoryError ……etc

2. It is possible to handle the exceptions by using try-catch blocks or throws keyword but it is not possible to handle the errors.

3. Error is an un-checked type exception.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q56. What are the differnet ways to handle the exceptions in java?

Ans: There are two ways to handle the exceptions in java.

1) By using try-catch block.

2) By using throws keyword

Exception handling by using try–catch blocks: To write the handling Code.

```
class Test
{       public static void main(String[] args)
        {       try
                {       System.out.println(10/0);
                }
                catch(ArithmeticException e)
                {       System.out.println("Exception raised....this is handling
code..."+(10/2));
                }
        }
}
```

Exception handling by using throws blocks: Used to throws the reponsibility of exception handling to JVM.

```
import java.io.*;
class  Test
{       static void m2() throws FileNotFoundException
        {       FileInputStream fis = new FileInputStream("marks.txt");
        }
        static void m1()
        {       try{Test.m2();}
                catch(FileNotFoundException e)
                {       System.out.println("File is not present....");
                }
        }
        public static void main(String[] args)
        {       Test.m1();
        }
}
```

Q57. What are the possible ways to handle the multiple exceptions in java?

Ans: Below all cases can handle multiple exceptions using single catch,

1. *catch(Exception e){}*
2. *using pipe symbol, catch(ArithmeticException | ClassCastException a)*
3. *using multiple catch blocks we can handle.*

(when we write the multiple catch block the catch block order should be child to parent,If we write the parent to child we will get error.)

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q58. What is the purpose of try-with resources concept?

Ans: a.When we declare the resources using try block, once the try block is completed the resource is automatically released.

syntax: try(Resource-1;resource-2;.....resource-n){

}

b. How it is release automatically means,every resource internally implemeting the AutoCloseable & it contains close() method to close the resources.

```
public interface java.lang.AutoCloseable {
        public abstract void close() throws
java.lang.Exception;
}
```

c. The close() method is invoked automatically on objects managed by the try-with-resources statement.

```
import java.util.Scanner;
class Test
{       public static void main(String[] args)
        {       try(Scanner s = new Scanner(System.in))
                {       System.out.println("enter id");
                        int a = s.nextInt();
                        System.out.println("input value="+a);
                }
        }
}
```

Q59. What is the purpose of finally block?

Ans: finally block is always executed irrespective of try-catch blocks.

finally block executed both normal & abnormal termination of application.

finally blocks is used to release the resource like,

database closeing,file closeing in both normal cases & abnormal cases.

```
try:
        connection open
        tx1
        tx2
        tx3
except ZeroDivisionError as e:
        sdflskdfjl
close the connection
```

1. All operations are success : normal termination : Connection closed

2. In tx2 exception raised : catch is matched : normal terminaton : Connection closed

3. In tx3 exception raised : catch is not matched : abnormal terminatio: connection not closed.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q60. What is the purpose of throw keyword? How to handle the userdefined exceptions in java?

Ans: Throw keyword used to throw the exception.

throw new ArithmeticException("your not eligible to marriage try after sometime...");

In above case we are throwing predefiend exception which is not recommended. Because the predefined exceptions are already having some fixed meaning.

step 1: creating user defined exception

```
class InvalidAgeException extends RuntimeException
{       InvalidAgeException(String msg)
        {       super(msg);
        }
}
```

step 2: Throwing user defined exception.

```
throw new InvalidAgeException("Your not eligible try after some time....");
```

Note: using throw keyword we can throw predefined exceptions & user defined exception.

But throwing predefined exceptions are not recommended because predefined exceptions are having fixed meaning.

Q61.What are the difference ways to create String object?

Ans: It is possible to create String object in two ways.

1) *Without using new operator*

```
String str = "ratan";
```

2) *By using new operator*

```
String str = new String("ratan");
```

#Creating a string obj without using new operator:

```
String str1 = "ratan";
String str2 = "durga";
String str3 = "ratan";
```

The objects are created in SCP(string constant pool)

In String constant pool memory just before object creation it is always checking previous objects.

a. If the previous object is not available it will create the new object.

b. If the previous object is available with the same content then the reference variable is pointing to existing object.

str1,str3 ----- ratan

str2 ----- durga

SCP does not allow duplicate objects.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

#Creating a string object by using new operator

```
String str1 = new String("ratan");
    String str2 = new String("durga");
    String str3 = new String("ratan");
```

The objects are created in heap memory

When we create object in Heap area instead of checking previous objects it directly creates new objects.

```
str1 --- ratan
str2 --- durga
str3 --- ratan
```

Q62. what is the difference == operator & equals() method?

Ans: == : operator always used for reference- comparison.

```
10==10  : True
String s1 = new String("ratan");
String s2 = new String("ratan");
System.out.println(s1==s2);          // False
```

equals() method present in object class used for reference comparison it return Boolean value.

If two reference variables are pointing to same object returns true otherwise false.

String is child class of Object and it is overriding equals() methods used to perform data comparison. If two objects data is same then returns true otherwise false.

StringBuffer class there is no equals() method it uses parent class(Object) equals() method used for reference comparison.

```
String s1 = new String("ratan");
String s2 = new String("ratan");
System.out.println(s1.equals(s2));
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q63. What is the difference between String vs. StringBuffer?

Ans:

String:

1. String is immutale once we create the object, modifications are not allowed

If the application requirment is to store the fixed data then use String class.

2. Two ways to create object

a. without using new operator

String str = "ratan"

b. by using new operator

String str = new String("ratan");

3. If the application requirement is to perform Data comparision use String class.

String class equals() method perform content comparision.

StringBuffer:

1. StringBuffer is mutable modifications are allowed.

If the applcation requirement to store the data with flexibility of modification then use StringBuffer class.

2. One way to create the StringBuffer using new

StringBuffer sb = new StringBuffer("ratan");

3. If the application requirement is reference-comparision then use StringBuffer class.

data comparision not possible.

Q64. What is the difference between equals() vs. compareTo()?

Ans:

equals() method will compare the data returns Boolean value.

If the both String are equals return true otherwise false.

"ratan".equals("ratan") = true

"ratan".equals("anu") = false

"ratan".equals("RATAN") = false

"ratan".equalsIgnoreCase("RATAN") = True

compareTo() method will compare the returns int value.

Compares two strings lexicographically. The comparison is based on the Unicode value of each character in the strings.

If the both String are equals return 0, otherwise returns positive or negative numbers. unicode numbers : a=97,b=98.... A=65,B=66....

"ratan".compareTo("ratan") = 0

"ratan".compareTo("anu") = 17 +ve

if the first string unicode value is bigger than second string unicode value then it return positive value. "anu".compareTo("ratan") = -17 -ve

if the first string unicode value is less than second string unicode value then it returns negative value.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q65. What is the difference between StringBuffer & StringBuilder?

Ans: StringBuffer(java 1.0) vs. StringBuilder(java 5.0):

StringBuffer:

A thread-safe(StringBuffer methods are synchronized so one thread can access at a time),mutable sequence of characters.

A StringBuffer is like a String, but can be modified.

At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.

StringBuilder :

A mutable sequence of characters. But with no guarantee of synchronization(StringBuilder methods can be accessed by multiple threads at a time).

This class is designed for use as a drop-in replacement for StringBuffer in places where the string buffer was being used by a single thread. Where possible, it is recommended that this class be used in preference to StringBuffer as it will be faster under most implementations.

Q66. What are the advanatges of Wrapper classes?

Ans:

Wrapper classes wrap the primitive data type into objects.

Java is an Object oriented programming language so represent everything in the form of the object, but java supports 8-primitive data types these all are not part of the object.

To represent 8-primitive data types in the form of object we required 8-java classes these classes are called wrapper classes.

All Wrapper classes are immutable classes present in java.lang package.

Advantages of wrapper classes :

a. Collections allowed only object data.

b. On object data we can call multiple methods. compareTo() equals() toString()

c. Cloening process only object data possible.

d. Object data allowed null values.

e. Serialization only object data....etc

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q67. Define Autoboxing & Auto-unboxing?

Ans:

Autoboxing:

Automatic conversion of primitive to wrapper object is called autoboxing.
The Autoboxing is done compiler.
The Autoboxing internally uses valueOf() method to create the wrapper object.

Auto-unboxing:

Automatic conversion of wrapper object to primitive is called auto-unboxing.
The Auto-unboxing is done compiler.
The Auto-unboxing internally uses xxxValue() method to get the primitive data.

```
class Test
{       public static void main(String[] args)
        {
//Autoboxing : Automatic conversion of primitive to wrapper object : valueOf()
                Integer num1 = 100;
                System.out.println(num1);

//Auto-Unboxing : Automatic conversion of wrapper object to primitive : xxxValue()
                int val = Integer.valueOf(100);
                System.out.println(val);
        }
}
```

Q68. What is the purpose of enum?

Ans:

enum is used to declare the group of constants.
In general the group of constants are public static final.
enums are introduced in java-5 verison.
Enums are by default final we can not create the child enums.

Before enum

```
class Week
{       public static final Week MON;
        public static final Week TUE;
        public static final Week WED;
}
```

java code with enum:

```
enum Week
{       MON,TUE,WED;   //public static final
}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q69. What is the purpose of Annotation?

Ans: Annotations are not to develop the application just it is giving the metadata information.

Annotations gives some metadata information to compiler.

Annotations starts with '@' symbol

General purpose annotation : programming level annotations : java.lang

@Override
@SuppressWarnings
@Deprecated
@FuntionalInterface
@SafeVararg

Meta annotation : annotation about annotations : java.lang.annotation

The meta annotations are giving information about general purpose annotations.

@Target
@Retention
@Documented
@Inherited

Note : meta annotations are giving information about general purpose annotations.

Q70. How to perform read & write operations on text files & images?

Ans:

To perform IO operations we need classes & interfaces these are present in java.io package.

Using java.io package classes & interfaces it is possible to work with only text files.

Java Application can read the data from text file & write the data to text file.

Channel is a communication medium to transfer the data.

Every Channel can do two operations,

a. read operations
b. write operations.

There are two types of channels,

1. Byte oriented channel: The data is transfer in the form of bytes : 8-bit : image data

FileInputStream : read the data
FileOutputStream : write the data

2. Character oriented Channel:The data is transfer in the form of charaters: 16-bit : text data

FileReader : To Read the data.
FileWriter : To write the data.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q71. What are the difference between normal Streams & buffered streams?

Ans:

Normal Stream :

FileReader : new FileReader("abc.txt")
FileWriter : new FileWriter("xyz.txt")
FileInputStream : new FileInputStream("Desert.jpg");
FileOutputStream : new FileOutputStream("ratan.jpg");

a. Reads the data char by char , If the file contains more characters it required more read & write operations it will effects on performance.
b. Normal streams interact with HardDisk to read the data.

BufferedStreams: these are developed based on normal streams

BufferedReader : new BufferedReader(new FileReader("abc.txt"))
BufferedWriter : new BufferedWriter(new FileWriter("anu.txt"))
BufferedInputStream: new BufferedInputStream(new FileInputStream("ram.jpg"))
BufferedOutputStream : new BufferedOutputStream(new FileOutputStream("Quotes.jpg"))

a. Reads the data line by line format, it improves the performance.
b. first time read the data from hard disk it will store into buffered memory, Later if we need the data it will read from buffered memory.

Q72. How to create & remove the files & directories in java?

Ans:

boolean java.io.File.createNewFile() throws IOException

Atomically creates a new, empty file named by this abstract path name if and only if a file with this name does not yet exist.

true if the named file does not exist and was successfully created;
false if the named file already exists

```
File file = new File("ratan.txt");
System.out.println(file.exists());
boolean status =  file.createNewFile();
```

boolean java.io.File.mkdir()

Creates the directory named by this abstract pathname.
true if and only if the directory wascreated; false otherwise.

```
File f = new File("usman");
boolean b = f.mkdir();
```

Note: To create the file under the directory the directory should be present.

```
File ff = new File("usman","ratan.txt");
ff.createNewFile();
```

To delete the file use delete() method.

```
File file = new File("ramu.txt");
boolean bb = file.delete();
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q73. what are the advantages of nested classes in java?

Ans:Declaring the class inside the another class is called nested class.

Nested classes are introduced in java1.2 version

There are two types of nested classes

1. Static nested classes

2. Non-static nested classes(Inner classes)

a. Normal inner.

b. Method local inner.

c. Anonymous inner.

a. we are grouping all logics in single place. encapsulating the data.

b. The inner class, can access outer class private properties.

c. readbility improved.

d. The main objective of inner classes is "One time usage...".

e. One functionality can exist with presense of another functionality then use inner classes.

Q74. Define lambda expression in java?

Ans: *To reduce the byte code & length of the code use lambda.*

(Arguments_list) -> expression

The lambda expression will provide implementations of only functional interface methods.

The lambda can take any

The advantages of lambda expressions:

a. It used to write the functional programming

b. It eliminates the boiler plate code.

c. It reduces the byte code.

d. It reduces the length of the code.

e. It improves the performance of the application.

```
Message m1 = ()->System.out.println("Good morning...Lambda");
Operations op1 = (int num1,int num2)->System.out.println(num1+num2);
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q75. What is method reference concept in java?

Ans: method reference concept in Java-8 version.

Method reference is used to refer method of functional interface. It is a short form of lambda expression.

Method reference will decide the varaibles,

Code with lambda :

```
list.forEach(emp->System.out.println(emp.getEid+" "+emp.getEname())):
Collections.sort(students,(s1,s2)->s1.rollno.compareTo(s2.rollno));
```

code with method reference:

```
list.forEach(System.out::println):
```

```
Collections.sort(students,Comparator.comparing(Student::getRollno));
```

Q76. What is Reflection mechanism in java?

Ans:

The process of gathering complete information about the class is called reflection.

In Reflection mechanism, the complete class information is stored in Class object.

The reflection classes, such as Method,Field,Constructor are found in java.lang.reflect.

To get the information about Demo class load the class first. The loaded class information stored in Class object.

```
Class c = Class.forName("Demo");

Field[] fields  = c.getDeclaredFields();

Method m[] = c.getDeclaredMethods();

Constructor[] constructors = c.getDeclaredConstructors();
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q77. What are the main components of JVM?

Ans: The JVM mainly contains 4-components,

1. class loader sub system

Used to loads the .class file byte code into memory.

2. runtime memory area

Method area : static data stored here

Heap area : instance data stored here

Java stacks : All local data

Pc(program counter) : next instruction to be executed

native method stacks : native methods information

3. execution engine.

used to execute the code.

4. Native method interface.

Used to interact with the native method libraries.

Q78. Define the Thread? What are the advantages of multithreading?

Ans: A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.

Thread vs. process:

a. Threads are light weight tasks created by JVM.

Process is heavy weight. created by Operating System.

b. One process contains multiple threads every thread is independent thread.

ex: Browser is a process & every tab is called thread.

Application areas of multithreading:

Server side implemented multithreaded environment, Because the server need to handle the multiple requests at a time. Every request is thread.

The main objective of multithreading is to develop the gaming application. Where the multiple objects are moving at a time.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q79. What are the different ways to create a thread? Explain start() run() methods?

Ans: we can create the thread in two ways,

b. extends Thread

ii. implements Runnable

```
class MyThread extends Thread
{       public void run()
        {       Logics Here…
        }
}
MyThread t = new MyThread();
t.start();

class MyRunnable implements Runnable
{       public void run()
        {       Logics Here…
        }
}
```

run() method to write the logics of the thread.

Start() method to start the thread.

The best way to create a Thread is By implements Runnable interface.

a. here it is possible to extends any other classes.

b. Here possible to apply lambda expression.

Q80. What's the difference between class lock and object lock?

And:Class Lock:

In java, each and every class has a unique lock usually referred to as a class level lock. These locks are achieved using the keyword 'static synchronized' and can be used to make static data thread-safe.

It is generally used when one wants to prevent multiple threads from entering a synchronized block.

```
public class ClassLevelLockEx
{  void classLevelLockMethod()
        {  synchronized (ClassLevelLockEx.class)
          {  //Logics Here
          }
        }
}
```

Object Lock:

In java, each and every object has a unique lock usually referred to as an object-level lock. These locks are achieved using the keyword 'synchronized' and can be used to protect non-static data. It is generally used when one wants to synchronize a non-static method or block so that only the thread will be able to execute the code block on a given instance of the class.

```
public class ObjectLevelLockEx
{ public void objectLevelLockMethod()
 {  synchronized (this)
        {  //Logics Here
   }
 }
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

}

Q81. Explain Daemon & non-Daemon threads in java?

Ans:

User Thread (Non-Daemon Thread):

In Java, user threads have a specific life cycle and its life is independent of any other thread. JVM (Java Virtual Machine) waits for any of the user threads to complete its tasks before terminating it. When user threads are finished, JVM terminates the whole program along with associated daemon threads.

Daemon Thread:

In Java, daemon threads are basically referred to as a service provider that provides services and support to user threads.

The threads are executing at back group to give the support to foreground threads.

ex: garbage Collector. (while JVM running Garbage Collector running)

Once the main/user thread completes execution all daemon threads are automatically stopped.

To set the daemon nature use setDaemon() method.

thread.setDaemon(true);

Q82. Explain thread names & priority in java?

Ans:

The default name of main thread is: Main

The default name of unedified threads : Thread-0, Thread-1, Thread-2....etc

In java every thread is having priority, the priority range 1-10

1- low priority 10-high priority

The default priority of threads : 5

If we set the priority more than 10 we will get IllegalArgumentException. If the application contains more than one thread with same priority then thread execution decide by ThreadSchedular.We need to set the priority before starting the thread.

Q83. Explain sleep() vs. join() vs. wait()?

Ans: All three methods are Overloaded methods used to stop the thread execution.

sleep():

Causes the currently executing thread to sleep (temporarily cease execution) for the specified number of milliseconds.

join() :

Waits for this thread to die. Remaining threads will wait unitl this thread completion. to stop other threads until our thread die use join().

wait():

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.

Q84. What is the purpose of Assertions in java?

Ans:

assert statements are used to debug the applciation by placing check points.

Debugging : identify the erros rectify the errors

An assertion is a statement in Java which ensures the correctness of any assumptions which have been done in the program.

Assertion are introduced in java-5 version.

syntax : assert condition:error_message

The assert statements are by default desable.To enable the assert statemetns use -ea (enable assertion)

if the assert condition is true then applcation is normal execution

if the assert is false we will get AssertionError with message.

When we get AssertionError think that we are putting some condition but it is fail.

Q85. Define the array? what are the advantages of arrays?

Ans:

Arrays are used to represent group of elements as a single entity & these elements are homogeneous & fixed size.

The size of Array is fixed it means once we created Array it is not possible to increase and decrease the size.

Arrays performance is good.

Array in java is index based first element of the array stored at 0 index.

By using arrays possible to store primitive data & object data.

```
int[] a = new int[6];
Emp[] e = new Emp[10];
```

If we know size in advance it is recommended to use arrays in application.

The only limitation about array is there is no methods, So operations becomes complex.

Q86. What is the main objective of Collection framework? What are the parts in Collections f/w?

Ans:

a.The main objective of collections framework is to represent group of objects as a single entity.

b. java Collection framework provide very good architecture to store and manipulate the group of objects.

c. Collection contains group of classes and interfaces that makes it easier to handle group of objects.

Collection framework mainly divided into two parts

a. Collection : used to store the data in single object format [10,20,30,40]

List implementation classes : ArrayList, LinkedList , Vector, Stack

Set implementation classes : HashSet, LinkedHashSet, TreeSet

Queue implementation classes: PriorityQueue

b. Map : used to store the data two objects format in the form of key:value

{"ratan":5,"anu":2,"sravya":10,"that":5}

Map implementation classes : HashMap , LinkedHashMap , TreeMap , Hashtable

Q87. What are the difference between Collections & Arrays?

Ans: Arrays vs. Collections:

Both Arrays and Collections are used to represent group of objects as a single entity but the differences are as shown below.

Arrays:

store the group of objects : homogenous

arrays are fixed in size : we can not increase & decrease the size

memory : int[] 100 : memory wise bad

Arrays are no methods : operations are complex

Arrays can store primitive int[] & it can store object data Emp[]

Collections:

store the group of objects : homogenous & heterogenous

Collections are growable in nature : size will increase & decrease

memory wise it is good

collections support methods operations are easy.

collections can store only object data.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q88. What are the difference between List & Set?

Ans:

public interface List<E> extends Collection<E>

lists typically allow duplicate elements.

An ordered collection (also known as a sequence).

If the application requirement is to store the duplicate element use List Classes,

The List implementation classes are,

ArrayList,LinkedList, Vector , Stack

public interface Set<E> extends Collection<E>

A collection that contains no duplicate elements.

An un-ordered collection.

If the application requirement is to store the unique element use Set Classes,

The Set implementation classes are,

HashSet, LinkedHashSet , TreeSet

Q89. How to add one collection data into another?

(or)

What is the difference between add() & addAll() method?

Ans: There are two ways to add one collection data into another collection.

i. By using constructor approach. : one to one [a1 is added in a2]

```
ArrayList<String> a1 = new ArrayList<String>();
a1.add("ratan");
ArrayList<String> a2 = new ArrayList<String>(a1);
a2.add("rani");
System.out.println(a2);
```

ii. By using addAll() method. : many to one [b1,b2 is added in b3]

```
ArrayList<Integer> objs1 = new ArrayList<Integer>();
objs1.add(20);
ArrayList<Integer> objs2 = new ArrayList<Integer>();
objs2.add(200);
ArrayList<Integer> objs3 = new ArrayList<Integer>();
objs3.addAll(objs1);
objs3.addAll(objs2);
objs3.add(1000);
System.out.println(objs3);
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q90. What are the Differences between ArrayList & LinkedList?

Ans: ArrayList vs. LinkedList:

1. ArrayList internally uses resizable array Data structure to store the elements.
So the data stored in array format.
LinkedList internally uses double linkedlist implementation to store the elements.
So the data stored in double linked list format.

2. when we are doing Insertion, remove data on Arraylist is slow because when we perform these operations internally it requires more shift operations.
But insertion,remove operations LinkedList is faster because shift operations are not required. Just adding or removing one link.

3. search operartions ArrayList is good it is index based.
Element retrieval in ArrayList is much faster then LinkedList, ArrayList take O(1) of time to retrieve any location where LinkedList take O(n) time to retrieve the element.
However, unlike arrays which allow random access to the elements contained within them, a link list only allows sequential access to its elements.

4. with respect to memory ArrayList is Good . it stores only data.
LinkedList stores the data & next link address.
Linked lists also use more storage space in a computer memory as each node in the list contains both a data item and a reference to the next node.

5. ArrayList is implements RandomAccess marker interface hence the data Access fast.
But LinkedList not implementing RandomAccess interface hence the data Access slow.

Q91. What are the Differeent ways to read the data from collection classes?

Ans: we can read the data from collection classes in different way,

1. *using for-each loop.*
2. *using get method.*
3. *using cursors*
   a. *Enumeration*
   b. *Iterator*
   c. *ListIterator*
4. *using forEach() method.*

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q92. What is the purpose of cursor? Explain the cursors in java?

Ans: To read the data from collection classes use Cursors, there are 3-types of cursors.

a. Enumeration
b. Iterator
c. ListIterator

1. Enumeration(i): java 1.0

Using this we can read the data from only legacy classes(java 1.0).
So it is not a universal cursor.
ex: Vector,Stack
Using this cursor we can do only read operations.
Using this cursor we can read the data only forward direction.

2. Iterator java 1.2 version

Using this cursor we can read the data from all collection classes, so it is universal.
Using this cursor we can do read & remove operations.
Using this cursor we can read the data only forward direction.

3. ListIterator(i) : java 1.2 version

Using this cursor we can read the data from only List implmentation classes,so it is not a universal
Using this cursor we can do read,remove,update,add operations.
Using this cursor we can read the data forward & backword direction.

Q93. What are the conditions we need to fallow while sorting Collection data?

Ans: if we want to perform sorting data it must satisfies the below conditions,

a. The data must be homogenous.
b. Must implements Comparable interface.

In java String,all wrapper classes are implementing comparable interface by default.
The Collection class methods internally fallows compareTo() method to perform sorting.
Different ways to perform sorting,

```
Collections.sort();
list.sort()
stream.sort()
```

case 1: if we are trying to perform sorting of heterogeneous data ,while performing comparison by using comapreTo() method JVM will generate java.lang.ClassCastException.
Using compareTo() not possible to compare two different objects.

```
ArrayList al = new ArrayList();
al.add("ratan");
al.add(10);
Collections.sort(al);
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

case 2: when we perform sorting of data, if the data contains null value while performing comparison by using compareTo() method JVM will generate java.lang.NullPointerException.

Any object with comparision of null, you will get NullPointerException.

```
ArrayList al = new ArrayList();
al.add("ratan");
al.add(null);
Collections.sort(al);
```

Q94. what is the purpose of serialization & Desrialization process? How to prevent serialization?

Ans: with in the JVM instance just public modifier is sufficient to get the permission.

JVM1 :module-1 module-2

If the project is running multiple JVM instances then public modifier is not sufficient to access the data, in this case use serialization & desrialization.

JVM-1 n/w JVM-2
m1 file m2

Serialization:

The process of converting java object into network supported form or file supported form is called serialization. (OR) The process of saving an object to a file is called serialization.

(OR)The process of writing the object to file is called serialization.

To do the serialization we required fallowing classes

1. *FileOutputStream*
2. *ObjectOutputStream*

Deserialization:

The process of reading the object from file is called deserialization.

We can achieve the deserialization by using fallowing classes.

1. *FileInputStream*
2. *ObjectInputStream*

The perform serialization our class must implements Serializable interfaces.

To prevent the serialization declare the property with transient modifier.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q95. What is Type Earser in java?

Ans: Type erasure:

The generic type information is available only at compile time but not at runtime.

The compiler will verify generic types and then removes the type information.

Hence, none of the type information is available at runtime. At runtime, both non-generic and generic code looks exactly same.

Example:

before compilation :

List<String> names = new ArrayList<String>();

After compilation:

List names = new ArrayList(); // type information is removed.

Hence generics are meant for compile time protection but not runtime protection since generic type information is not available at runtime.

Q96. What is the Difference between Comparable vs. Comparator?

Ans:

if we want to perform sorting data it must satisfies the below conditions,

a. The data must be homogenous.

b. Must implements Comparable interface.

The String & wrapper classes are implements Comparable Interface.

java.lang.Comparable:

To perform default natural sorting we will use Comparable, only String & wrapper classes are implements Comparable Interface.

It gives compareTo() method technique to sort elements.

If you want to sort the elements as per the default natural sorting order, you must choose the Comparable interface.

java.util.Comparator:

It is used to perform the sorting of userdefiend classes like Emp,Student....etc. The predefiend classes are not implementing this interface.

It gives compare() method technique to sort elements.

The comparator interface helps to order the objects of the user-defined classes.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q97. In application when we will use Map classes?

Ans:

a. An object that maps keys to values.

A map can not contain duplicate keys; each key can map to at most one value.

b. Map is used to store two objects at a time in the form of key value pairs.

Here the key is object & value is object.

c. The key value pair is known as entry, the map contains group of entries.

d. In the map the keys must be unique but values we can duplicate.

1. To store the group of objects.

keys values

references objects

str1 ----- ratan

str2 ----- anu

str3 ----- ratan

2. To count the word occurence.

ratan ---- 5

anu ------ 2

is ----- 5

3. ip no of hits

10.5.6.4 ----- 5

10.5.4.2 ----- 10

78.90.45.4 ----- 5

4. To store the form data.

text filed logical name value

user name logical name ---- ratan

user pass logical name ---- ratan

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

Q98. What is the purpose of streams API in java?

Ans:

Introduced in Java 8, the Stream API is used to process collections of objects. A stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result. A stream is not a data structure instead it takes input from the Collections, Arrays or I/O channels.

Streams don't change the original data structure, they only provide the result as per the pipelined methods.

map: The map method is used to returns a stream consisting of the results of applying the given function to the elements of this stream.

```
List number = Arrays.asList(2,3,4,5);
List square = number.stream().map(x->x*x).collect(Collectors.toList());
```

filter: The filter method is used to select elements as per the Predicate passed as argument.

```
List names = Arrays.asList("ratan","anu","sravya");
List result = names.stream().filter(s-
>s.startsWith("S")).collect(Collectors.toList());
```

sorted: The sorted method is used to sort the stream.

```
List names = Arrays.asList("ratan","anu","sravya");
List result = names.stream().sorted().collect(Collectors.toList());
```

Q99. How to convert Arrays to Collections & Collections to Arrays?

Ans: Converting arrays to Collections & Collections to Arrays

1. *Arrays to collection : Arrays.asList()*
2. *collections to arrays*
   a. *toArray(arg) : generic version of collection to array.*
   b. *toArray() : normal version of collection to array.*

```
//conversion of arrays to collections
String[] str = {"ratan","anu","sravya"};
ArrayList<String> objs = new ArrayList<String>(Arrays.asList(str));
objs.add("aaa");
System.out.println(objs);
ArrayList<String> a1 = new ArrayList<String>();
a1.add("ratan");          a1.add("anu");
a1.add("sravya");
String[] s = new String[a1.size()];
a1.toArray(s);

//normal version collection to Array
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com

```
ArrayList data = new ArrayList();
data.add(10);          data.add("ratan");   data.add(10.5);
Object[] o = data.toArray();
```

Q100. What is the purpose of WildCard symbol(?) in Generics?

Array type information is available at runtime. Since JVM knows array type, hence, it throws
ArrayStoreException if we try to add different animal type. Arrays are polymorphic.

```
static void addAnimals(Animal[] animals)
  animals[0]   = new Dog();  // valid
```

Collections generics are non polymorphic because the generic data remmoves at compile time.

```
void addAnimals(List<Animal> animals){
}
List<Dog> dogs = new ArrayList<Dog>();
addAnimals(dogs);//C.E
```

Wildcard (?) symbol supports generic type polymorphism in java.
1. <? extends generic-type> : here we can do only read operataion but not add operation.
The wildcard (?) symbol with 'extends' guarantees that , just we use collection for retrieving elements but not for adding new elements. We can send any generic type which is subtype of generic supertype.

2) <? super generic-type> : here we can read & add
This is used to add elements. Accept generic type which is same or any one of its super type, i.e., anything higher in the inheritance hierarchy, but not lower in the inheritance hierarchy.

3) <?> : it can take all types.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 88 85 25 26 27, 72 07 21 24 27/28 | www.durgasoftonline.com
Maii: durgasoftonline@gmail.com