1) What is mean by data type?

- Data types are used to create the container and also it will specify the size and size if the data.
  There are two types-
  a) Primitype datatype :
  -It is used to store single valued data.
  -byte, short, int, long, float, double, char, boolean
  b) Non primitive datatype:
  -It is used to store multi valued data
  -String, arrays, interface, class, package

2) What is mean by variable?

- Variables are used to store the data for future use.

-Based on scope of visibility variables are classified into two types.

a) Local Variables

b) Global variables

-Global variables are again classified into two types:

1. Static Global variable

2. Non- Static Global Variable

3) What is mean by typecasting?

- The process of converting one type of data into another type of data is known as typecasting.

1. Primitive type casting:

The process of converting one primitive data type into another primitive data type is known as primitive typecasting.

Widening: The process of converting smaller data type into larger data type is known as widening. There is no any data loss in widening so compiler going to do it implicitly. It is also called as auto widening.

Narrowing: The process of converting larger data type into smaller data type is known as narrowing. There is no any data type so compiler will not do it implicitly. The programmer has to do it implicitly.

1. Non primitive type casting:
   The process of converting one non primitive data type into another non primitive data type is known as non-primitive type casting.
   The process of converting one type of reference to another type of reference is known as non-primitive typecasting.
   Up casting: It is the process of converting sub class reference into superclass reference type. (Up casting is done implicitly)
   Down casting: It is the process of converting super class reference into sub class reference type.
   (Programmer has to do down casting explicitly)

4) What is mean by Operators?

Operators / symbols which has specific operation to be done.

Based on number of operands which an operator accepts, it is classified into three types

1. Unary operators:
   Unary operator accepts only single operand
   Ex. Typecast operator, Logical NOT operator

2. Binary operators:
   Binary operator accepts two operands
   Ex. Arithmetical operators, Relational operators, Logical operators

3. Ternary Operators:
   An operator which accepts three operands is known as ternary operators.
   Ex. Conditional operator

5) What is Decision Making Statement?

Whenever we want to make any decision based on condition then we use Decision making statements. Decision making statements are used to skip set of Instructions based on condition.

In java we have following decision making statements

1. if statement (When we have only one condition)
2. if else statement (When we have exactly two conditions)
3. else if ladder (When we have more than two conditions)
4. switch (When we have to compare values with different cases)

6) What is break?

Break is a keyword.
Break is a control transfer statement; it transfers the control outside of the block.
Break can be used inside the switch and loops.

7) What is loop?

Whenever set of instruction or same set of statements are repeated multiple time then we use loops.

Below things are taken considering while using loop

1. Initialization
2. Condition
3. Updation

In java we have following loop statements:

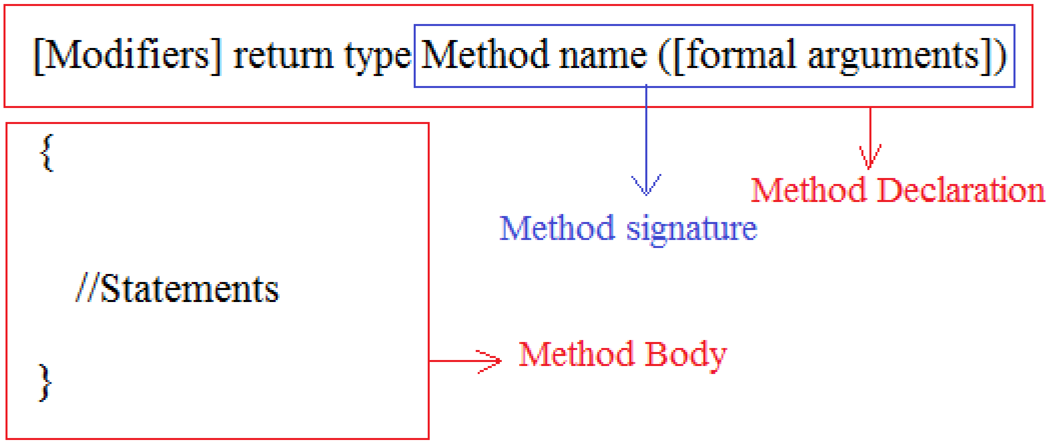
1. While loop
2. do while loop
3. for loop
4. for each loop/ Advanced for loop / enhanced for loop

8) What is method?

Method is a block of instruction to perform specific task.

Methods will get execute only when it is called.

Syntax:



```
[Modifiers] return type Method name ([formal arguments])
{

    //Statements

}
```

Method Defination: Method declaration along with method implementation is called as method defination.

There are two types of methods –

1. No Argument Method: A method which will not accept any argument is known as no argument method.
2. Parameterized Method: A method which is declared with formal arguments is known as parameterized method.

9) What is return?

Return is a keyword.

It is control transfer.

It is used to transfer the control from called method to calling method.

We should give return statement as the last statement of the block.

If there is any statement within the block, we will get CTE as unreachable statement.

Return type: The type of value return to its caller is known as return type.

We can have following return types for method:

1) Primitive types
2) Non Primitive types
3) Void:
    a. Void is a keyword
    b. It represents data type (Nothing)
    c. If the return type of method is void it is not mandatory to use return statement.

9) What is method overloading?

- The process of creating more than one method with a same name but different formal arguments is known as method overloading.

-The formal arguments must be differing either in length or in type or in the order of declaration.

-Return type can be anything

-Always the highest priority will go for same type.

Compiler achieves the binding between method call statement and method implementation to be executed at compile time based on actual arguments passed this is known as compile time binding.

10) Method Recursion:

Method calling itself is known as method recursion. It is classified into two types:

1. Direct recursion: Method calling itself is known as direct method recursion.
2. Indirect recursion: Two methods calling itself is known as indirect recursion.

11) What is static and static members?

- Static is keyword

- Static is a modifier

- Any member of a class prefix with static modifier is known as class member / static members.

A. Static Methods:

- A method prefixed with static modifier is known as static method

- A static method is also known as class method

- The reference of static method is stored inside class static area

- The static method block is also known as static context.

Static methods can be called in two ways within the same class.

a. Directly with the help of method name.
b. With the help of class name as a reference.
c. A static method of a class can be used inside a member of another class with the help of class name to which method belongs to.

B. Static Variable:

- A global variable which is prefixed with static modifier is known as static variable.

- A local variable will have default values

* Static variable's memory is allocated in class static area hence it is also known as class variable .

* A static variable will be assigned with default value hence it can be used without initialization.

* A static variables can be used within the same class directly or with the help of class name.

C. Static initializer: We have two static initializers -

a. Static declare and initialization statement

b. Static block

* The static initializers get executed during the loading process of class (that is before main method is called)

* Static initializers get executed only once for each time when class is loaded.

* Static variables can be accessed with the help of object refrence.

12) Explain the loading process of a class

1. Class static area will be created for that particular class

2. All the method should load inside method area

3. If any static methods are present then those static method signature along with the reference (address) should load inside class static area.

4. If any static variable are present then, those variable should load with default values inside Class static area.

5. If any initializers are there then they get executed. Then the loading process of class is said to be completed.

13) What is non-static and non-static members?

- Any member of class prefixed without static modifier is known as non static member of a class.

- Non static member gets their memory allocation within the instance of a class (Object) Therefore we can use non static members from the static context with the help of address of an object.

-The process of creating an object for a class is known as instantiation.

- With the help of ‘new’ keyword we can create a block of memory in heap area. New operator returns the address of an object created. Every time new operator is used a new object is going to create and address will be different.

A. Non static variable:

* A global variable which is not prefixed with static modifier is known as non static variable.

* Non static variable will be allocated inside the object of a class which is created

* Non static member can not be used directly or with the help of class name inside static context.

* We can use the non static members from static context only with the help of reference. (Address of an object)

* Non static variable will have one copy each object of the class created, therefore if we created n number of objects for a class we will be having n copies of non static variable (One in each object)

B. Non static Methods:

* A method which is not prefixed with static modifier is known as non static method.

* The block which is belongs to non static method is known as non static context.

* We cannot call non static method from static context directly; we get CTE.

* Any block which belongs to non-static members is known as non static context. Inside non static context we can use static members of class directly, We can use non static members of same class directly.

C. Non static initializers:

* Non static initializers are gets executed during loading process of an object.

* It will get execute for each and every object during its loading process.

* Non static initializers of two types:

1. Non static declare and initialization statement.

2. Non static block:

A non static initializer (block) gets executed only once for each object.

*We can use non static block to count the number of objects created for a class.

14) What is ‘this’?

1. this is a keyword

2. his is a non static variable which will have the address of object currently used Address of currently used

3. this keyword can be used inside the non static context.

4. It is used to access the members of an object from a non static context.

5. If a local variable name and object variable name (non static variable name) is same we can use the object variable with the help of this keyword

15) What is constructor?

- Constructor is special non static member of a class which is used for object creation of a class (Instantiation of class)

- We have two types of constructors:

A. No argument constructor

* The no argument constructor added by compiler while compilation this known as default constructor.

* Compiler will add default constructor only when the constructor are not defined by the programmer.

PREDEFINED INSTRUCTIONS OF CONSTRUCTOR:

- The compiler would add instructions which helps to load the non static members of the class into object.

- The non static initializers written by programmer will added inside constructor body by the compiler.
- Constructors can be defined by programmer & it is known as user defined constructor.

B) Parameterized Constructor:

- A constructor which accepts arguments is known as parameterized constructor.
- Parameterized constructor are used to initialize the states of an object during the loading process (Object creation)

16) Explain the loading process of an object

1. 'new' creates block of memory in heap area

2. The constructor used in object creation is called.

3. Loading of non-static member of a class into the object takes place.

    a. Non static variables will be allocated with default value

    b. Reference of non-static method will be stored inside the object.

4. Non static initializers of class gets executed

5. The user written instructions of constructor gets executed once the execution of constructor is done, then the reference of object will be return. Then the loading process of object is said to be done.

17) What is constructor overloading and constructor chaining?

- A class having more than one constructor is known as constructor overloading.

- We can create n number of constructors within the same class but-

1. Name of the constructor must be same as class name

2. Constructor declaration must change in its declaration that is in the length or type or order of formal arguments.

Constructor chaining:

A constructor calling the another constructor is known as constructor chaining.

We can achieve constructor chaining in two ways:

A. **This call statement**: It is used to achieve constructor chaining within same class.

   Rules:

   1. This call statement can be used only inside a constructor
   2. It should always be the first statement inside constructor body
   3. This call statement should not call constructor recursively.

4. If a class has n number of constructors we can use this call statements only in n-1 number of constructors

B. **Super call statement:** It is used to call the constructor of superclass from subclass.

Rules:

1. super call statement can be used only inside a constructor
2. It should always be the first statement inside constructor body
3. We cannot use both super() and this() in one constructor.

PART: 2

1) What is Encapsulation?

* Encapsulation is the process of binding the states and behaviors of an object.

*In java, We achieve encapsulation with the help of CLASS.

*With the help of encapsulation we can achieve data hiding.

Data hiding:

Data hiding is the process of restricting the direct access to the states of an object and providing the controlled access with the help of behavior of the same object(class) is known as data hiding.

Steps to achieve Data hiding:

*Prefix the state of an object with the help of private access modifier.

*Define getter and setter method to access private data member.

2)When do we use getter and setter method?

Getter method:

*Getter method is used to fetch the private data member from a different class.

*Getter methods should returns the data which has to be read.

Setter method:

*Setter method is used to modify the private data member of a class from a different class.

*Setter method must accept an argument.

3) How do we make a field only readable?

a) Make a data member as private.

b) Create only getter method.

4) How do we make a field only modifieble?

a) Make a data member as private.

b) Create a setter method.

5)How do we achieve encapsulation in java with an example.

*In java, We achieve encapsulation with the help of class.

We represents the states of an object with the help of non-static variable and behavior of an object with the help of non-static methods.

EXAMPLE:

```
class Saving Account

{

int acno;

String name;

double balance;

String ifsc;

SavingAccount (){}

SavingAccount (int acno,String name,double balance, String ifsc)

{

this.acno=acno;

this.name=name;

this.balance=balance;

this.ifsc=ifsc;

}

public void withdraw(double amt)

{

      if(amt<=balance)

       {

          balance=balance-amt;

          sopln(amt+" is withdrawn succesfully");

          sopln("Remaining balance is "+balance);

      }

    else

     {

      sopln("INSUFFICIENT BALANCE");

     }

}

     public void deposit(double amt)
```

```
      {
         balance=balance+amt;
        sopln(amt+"  is deposited succesfullt");
        sopln("UPDATED BALANCE IS "+balance);
      }
       public void display()
         {
             sopln("Account Details");
             sopln("Account Number :"+acno);
             sopln("Account Holder Name :"+name);
             sopln("Account Balance :"+balance);
             sopln("Account IFSC :"+ifsc);
        }
}
//Driver class
class SavingAcountDriver
{
    public static void display(String[]args)
     {
         SavingAccount s1=newSavingAccount(123,"Dinga",30000,SBIN00041114);
        s1.display();
        s1.withdrawn(5000);
        s1.deposit(10000);
}
}
```

6)what is IS-A Relationship ?

Is A Relationship is similar to parent and child relationship where child acquires the properties and behaviours of parent.

Example:

*Car is a vehicle.

*QSPIDER is an institute.

*Peacock is a bird.

*Dog is an animal.

>>>> In java we achieve IS A relationship with the help of Inheritance concept.

7)How do we achieve Is A relationship in java, Give an example.

We can achieve IS-A relationship with the help of inheritance.

INHERITANCE:

>>Inheritance is a design techinque which is used to establish IS-A relationship,it is a process where one type of object acquires the properties and behaviours of another type of object.

PARENT CLASS:

>>The class which provides the states and behaviours to another class is known as parent class.

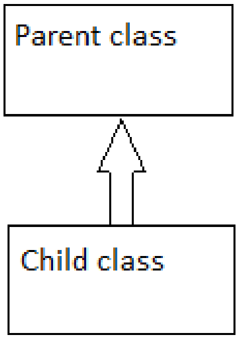
Parent class is also known as Super class/base class/Generalised class.

CHILD CLASS:

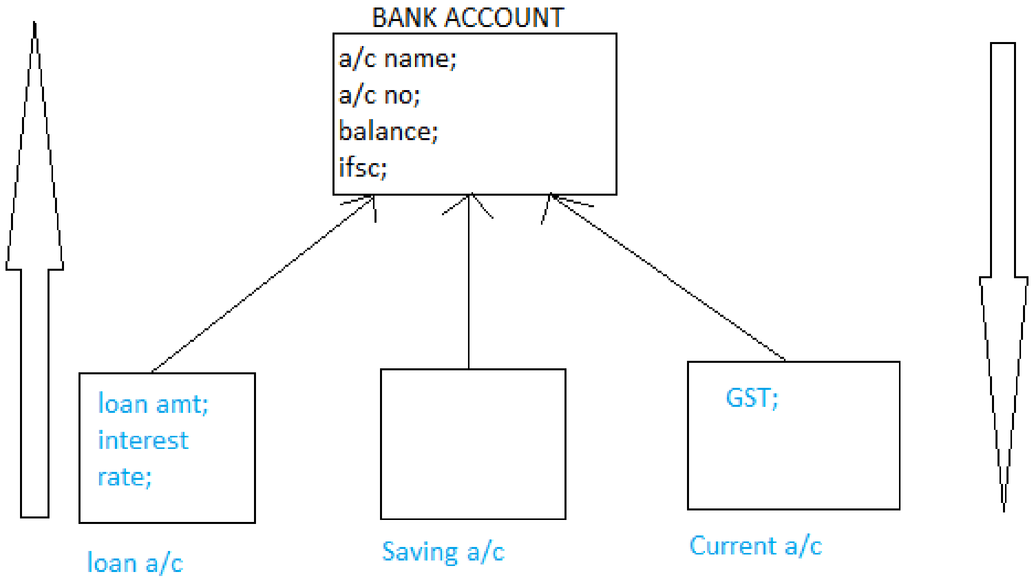
>>The class which recieves the states and behaviours from parent class is known as child class.

Child class is also known as Sub class/derived class/specialised class.

Inheritance can be represented as belows:



Arrow head represents the super class,and arrow tail represents the sub class.



Let us consider only accounts and loan accounts,Here loan a/c will have inherited members (members of a/c class) as well as declared in Loan a/c.Therefore the states of Loan accounts are

>a/c number

>a/c name

>balance

>ifsc code

>Loan a/c

>Interest rate.............

8)How do we achieve Inheritance b/w two classes in java. Explain with example.

In java,We achieve is A relation with help of keywords

*extends

*implements.

>Extends keyword is used to achieve inheritance b/w two classes.

>Extends keyword should be used in the SUB class.

SYNTAX:

```
class Parent

{

}

class Child extends Parent

{

}
```

>>A class can extends only one class .We cannot extend more than one class.

EXAMPLE:

```
class BankAccount
{
int acno;
String name;
double balance;
String ifsc;
BankAccount(){}
BankAccount(int acno,String name,double balance,String ifsc)
{
this.acno=acno;
this.name=name;
this.balance=balance;
this.ifsc=ifsc;
}
public void accountDetails()
  {
    sopln("Account Details");
    sopln("Account Number :"+acno);
    sopln("Account Holder Name :"+name);
    sopln("Account Balance :"+balance);
    sopln("Account IFSC :"+ifsc);
  }
}
//inheriting
class LoanAccount extends BankAccount
double loanamt;
double interest;
LoanAccount()
{}
LoanAccount(int acno,String name,double balance,String ifsc,double loanamt,double interest)
```

```
{
this.acno=acno;
this.name=name;
this.balance=balance;
this.ifsc=ifsc;
this.loanamt=loanamt;
this.interest=interest;
}
public void LoanAccountDetails()
  {
    sopln("Account Details");
    sopln("Account Number :"+acno);
    sopln("Account Holder Name :"+name);
    sopln("Account Balance :"+balance);
    sopln("Account IFSC :"+ifsc);
    sopln("Loan amount :"+loanamt);
    sopln("Loan Interest :"+interest);
  }
}
//Driver class
class BankDriver
{
public static void main(String[]args)
{
BankAccount b1=new BankAccount(123,"Dinga",35000,"SBIN00041114");
b1.accountDetails();
}
}
//Loan account
class LoanDriver
{
   public static void main(String[]args)
```

```
    {
        LoanAccount l1=new LoanAccount(456,"Dingi",45000,100000,7);
      l1.LoandAccountDetails();
     }
}
```

8) What is extends?

>> extends is a keyword.

>> extends is used to achieve inheritance between 2 classes.

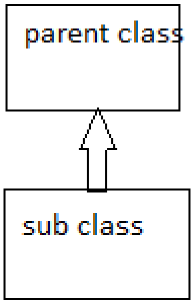
>>A class can extends to only one class. It cannot extends more than one class.

>>extends keyword should be used in SUB class.

9) What are the diffrent types of inheritance. Explain with an example.

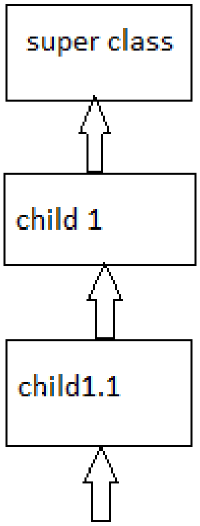
Single level Inheritance:

If the Inheritance is happening on single level is known as Single level Inheritance.



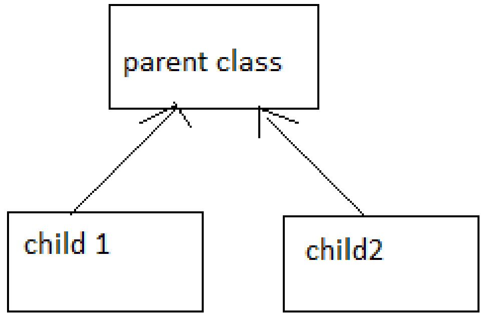
Multi Level Inheritance:

If the inheritance is happening in multi level is called s



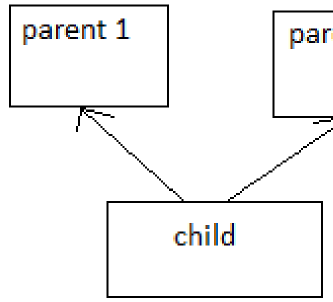
Hierarcheal Inheritance:

If a super class has more than one sub class in same level is known as Hierarcheal Inheritance.



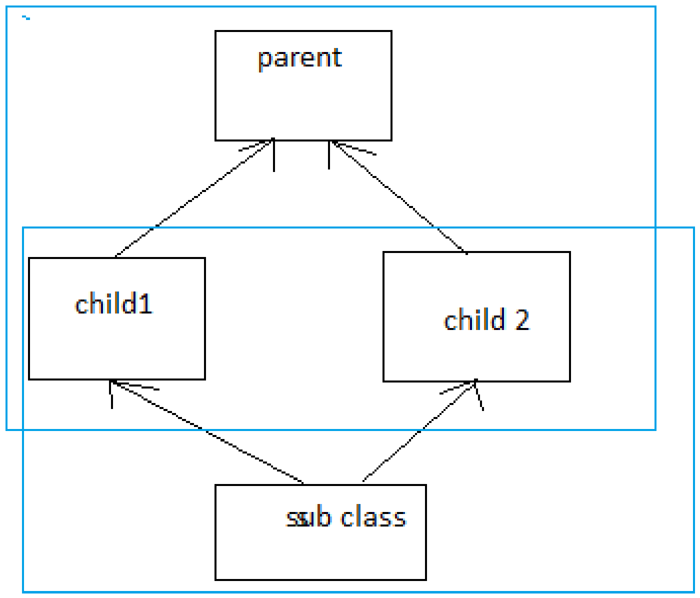
Multiple Inheritance:

A sub class having more than one super class at the same level is known as Multiple Inheritance.



Hybrid Inheritance:

it is a combination of Hirarcheal and Multiple Inheritance is known as Hybrid Inheritance.



10)Why cant we achieve multiple inheritance with the help of classes in java?

Multiple inheritance has an ambiguity problem is known as diamond problem.

Because of this diamond problem in java, we cant achieve multiple inheritance only with the help of classes.

A class can in java can inherit only with the help of classes.

Case 1:

>> Let us consider class C extends class A as well as class B.

>>When we create an object for class C,during the object loading process of object C,the constructor of class C tries to call the constructor of the super class,with the help of super() statement .

>>At this situation,there is an ambiguity problem to call either constructor of class A or contructor of class B.it is known as Diamond problem.

Case 2:

Let us consider class C extends to class A as well as class B

>>Assume both the superclass have methods with same signature,if we tries to call the method with the help of sub class referrence,there is an ambiguity weather to execute implementation of class A method or to execute the implementation of class B method.

>>It is also known as ambiguity problem.

11) Static members of a class are inherited or not.Justify with an example.

Static members of a class is inherited.

>>In the loading process of a class,parent class is loaded first and then the sub class or child class gets loaded.

>>Sub class static area will be have a referrence to the super class static area,Therefore

a)In sub class we can use static members of sub class directly(inherited).

b)With the help of sub class name,we can use static members of sub class as well as super class.

Example:

```
class A
{
static int a=10;
}
class B extends class A
{
    public static void main(string[]args)
       {
            sopln("MAIN BEGIN");//MAIN BEGIN
           sopln(a);//10
           sopln("MAIN END");//MAIN END
      }
}
```

12) Non static members of a class or inherited or not,Justify with an example.

Non static members of a class are inherited.

>>In the sub class we can use non-static members of super class.

>>With the help of sub class object refference,we can use the members of sub class as well as super class.

>>With the help of super class object refference We can use only the member of super class and we cannot use the members of sub class.

Loading process of object with respect to Inheritance.

>>When we create an object of sub class,the sub class object will have the instance of super class and instance of super class.

>>This is done with the help of constructor chaining using super call statement.

EXAMPLE:

```
class A

{

int a=10;

}

class B extends class A

{

        public void test()

       {

         sopln("a :"+a );

       }

}

Class Driver1

{

   Public static void main(String[]args)

       {

          B obj=new B();

          obj.test();

         sopln(obj.i);

      }

}
```

13) What is super () statement? why do we use it?

>>Super call statement is used to call the constructor of super class from the sub class constructor.

>>Super call statement will be implicitly added by the compiler to all the constructor of the class which doesnt have this() statement.

MODIFIERS AND PACKAGES:

14)What is package?

Package is a mechanism which is used to store group of similar java files together.

In java, we have two types:

a) Built in packages

b) User defined packages.

14) Why do we need packages?

A package in java is used to group related classes.It as a folder in a file directory.We use packages to avoid name conflicts,and to write better maintainablecode.

>>Java package is used to categorize the classes and interfaces so that they can be easily maintained.

>>Java package provides access protection.

>>Java package removes naming collision.

15) How to use member of one package in another package?

In java, we can only use a public member outside package. A non- public member cannot be used outside the package.

RULE:

>We can access public member outside a package in two ways:

1)Fully qualified name: The member name along with the package name is known as fully qualified name.

2) Import keyword: it is used to load a class from a specified package into the current package.

16)What is import?

>import is a keyword.

>It is used to load a class from specified package into the current package.

Ex: import packagename.classname;//for a particular class

Import package name.*;//for all the classes present in that package.

17)Can we use the members of a package outside a package without using import.?

Yes, we can access the public member outside a package without import keyword.

i.e., with the help of fully qualified name.

Ex:packagename.classname.member name;

18)What is modifier ,explain the types of modifiers?

Modifiers are the keywords used to modify the behaviors of java members.

We have two types of modifiers and they are:

>Access modifiers(Access specifiers)

>Non Access modifiers

Access Modifiers:

1)Private:

Private is an access modifier,any member of a class which is prefixed with private keyword is known as private member.

>>private member can be used only within same class.

>>It cannot be used in different class or sub class present in same package and it cannot be used in different class or sub class present outside the package.

2) Default/package scope:

If a member is not prefixed with any access modifier,then it is known as default scope or package scope.

>>A member with a package scope can be used within the same class or sub class or different class present in the same package.

>>It cannot be used in different class or sub class present outside a package.hence it is known as package scope.

3) Protected:

Protected is an access modifier, any member of a class which is prefixed with protected modifier is known as protected member.

>>It can be used within the same class, different class or sub class present inside a same package and it is also can be used in sub class of different package.

>>It cannot be used in a different class present outside a package.

Note:A class cannot make it as protected.

4)Public:

It is an access modifier.

>>Any member of a class which is prefixed with public keyword then it is known as public member.

>>It cannot be used in same class,different class,sub class of same package as well as in different package.

	Private	Default	Protected	Public
Same class	YES	YES	YES	YES
Different class of same package	NO	YES	YES	YES

Sub class of same package	NO	YES	YES	YES
Different class of different package	NO	NO	NO	YES
Sub class of different package	NO	NO	YES	YES

19) Can we make a constructor private?explain?

Yes, we can declare a constructor as private.But if create a constructor as private we are not able to create object of a class outside the class.

If we try to extend a class which is having private constructor, compile time error will occur. A private constructor doesn't allow a class to be sub classes.

POLYMORPHISM:

20)Explain Polymorphism ? and its types?

Polymorphism is the ability of an object to exhibit more than one form.

In java , we can achieve polymorphism in 2 ways

>>Compile Time Polymorphism(Static binding)

>>RunTime polymorphism(Non static binding)

Compile time polymorphism:

Compiler decides which implementation to be executed for method call statement is known as compile time polymorphism.

Compile time binding is done with the help of method signature. This is known as Compile time polymorphism.

1)Method overloading.

2)Constructor overloading.

3)Shadowing >Variable shadowing

>Method shadowing

Run Time Polymorphism:

It is also known as Non-static binding.In run time polymorphism,the bind between method call statement and method implementation happening during run time,hence it is known as run time polymorphism.

>>We can achieve run time polymorphism with the help of Method overriding.

21)Explain Shadowing?

Shadowing:

Variable shadowing:

If the super class and sub class having either static or non static variable with the same name,it is known as variable shadowing.

>>It is the compiler who decides which variable to be used based on

**Based on invocation/usage.

**Based on refference type.

>>If the referrence type is super class,super class member is used.

>>If the refference type is of sub class,sub class member is used.

>>It does't decide on the object reaction.

Method Shadowing:

If the super class and the sub class having the static methods with same signature,it is known as method signature.

Rules:

>>There should be is a relationship b/w two classes.

>>Both classes have static methods with same signature.

>>The return type of both the methods should be same.

>>The access modifier of sub class static method must be either same or greater visibility compare to super class static method's access modifier.

>>In method shadowing , the compiler decides which implementation to be executed based on

>>Place of invocation/usage.

>>Based on referrence type.

>>It doesn't depend on object reaction.

22)What is Overriding?

The process of providing an implementation to super class method with the help of sub class is known as Method overriding.

Rules to achieve method overriding:

>>Must have IS A relationship.

>>In the sub class,non static method signature should be same as super class non static method.

>>The return type of both methods should same.

>>The sub class access modifier should be either same or higher visibility compared to super class access modifier.

>>NOTE:Which method implementation to be executed will be decide on object creation,and not depend on referrence variable.

Static methods cant be overridden.

23)What is final modifier?

>>It is a keywod.

>>It is a modifier.

>>We can use final modifier for *Variables

*Methods

*Class

Final Variable:

>>A variable which is prefixed with final modifier is known as final variable.

>>A final variable behaves like a constant .

>>We can have 2 categories of final variable

*Local Final variable.

* Global Final variable.

Final Methods:

If a method is prefixed with final modifier then it is known as final method.

>>Final methods will be inherited..but cannot overriden.

Final class:

If any class is prefixed with final modifier,is know as final class.

>>Final class cannot be inherited.

24)Explain Abstraction?

It is one of the oop's design technique,which helps us to provide only the essential feautures of an object and hide their implementation details from the user.

With respect to abstraction,we can classify methods into 2 types

1)Concrete method

2)Abstract method

Concrete Method:

A method which has method body(implementation) is known as concrete method.

Abstract Method:

A method which does't have method body or implementation is known as abstract method.

Method declaration must be prefixed with abstract modifier

>>Method declaration must end with semicolon.

>>Method must not have body.

With respect to abstraction,we can classify class into 2 types

1)Abstract class

2)Concrete class.

Abstract class:

A class which is prefixed with abstract modifier is known as abstract class.

>>In abstract class will have constructor.

>>Abstract class can have concrete method as well as abstract method.

>>Abstract class cannot be instantiated.(We cannot create an object for abstract class)

Concrete class:

A class which is not prefixed with abstract modifier Is known as concrete method.

>>Abstract class can't have abstract method.

>>Concrete class can be instantiated.

Note:

>>Static methods cannot be made as abstract method.

>>We cannot make the abstract class as final.

>>We cannot make abstract method as final.

>>By using abstraction ,we can achieve only 0-100% abstraction.

25)When we should make a class as abstract?

If a class has atleast one abstract method either declared or inherited But not overriden

Then it is mandatory to make the class as abstract.

26)What is an Interface?

Interface is a non-primitive user defined datatype which is used to achieve 100% abstraction.

>>By default interface is abstract.

>>We cannot create an object for interface.

>>We cannot make an interface as private or protected,it can be either public or default.

Members of interface:

>>We can have non static abstract method.

>>We can have only public static final variables.

>>In interface , we can have public static methods.

Members we cannot have in interface:

>>Constructors

>>Non static concrete methods

>>Non static variables

>>Initialization blocks.

>>Non final static variables

>>Private methods

>>Protected methods

PART : 3

2) What is mean by array?

-Arrays is a continuous block of memory which is used to store homogeneous data/values.
-The values stored inside array is known as elements.
-In java array is an object.
-Array reference variable is used to store the address of array object.
-The default value of array reference variable is null. Therefore inside the reference variable of an array we can store either null or address of respective array objects.

-We can create an object of array in 2 ways

1) Declare and initiation statement

2) By creating the object using new operator

Yes, we can create empty array object. Which will have length as zero. We can find the length of the array with the help of non-static variable of array object.

-We can access elements of array with the help of index.

2) What are the types of array?

Multi-Dimensional array:

Multidimensional arrays are nothing but an array with more than one dimension ie more than one row or one column.

In multidimensional arrays we have basically 2 types

1) 2-Dimensional array
2) 3-Dimensional array

2-Dimensional Array:

2D arrays are represented as array of arrays. It can be considered as representation of matrix.

“Jagged Arrays”.

1) Jagged arrays are the arrays which consists of elements of different sizes in each array.
2) While declaring jagged arrays, it is not required to mention the size of secondary array.

3) What are the drawbacks of array in java?

To store multiple objects or group of objects together we can generally use arrays. But arrays has some limitations

Limitations of Array :

The size of the array is fixed, we cannot reduce or increase dynamically during the execution of the program.

Array is a collection of homogeneous elements.

Array manipulation such as :

- removing an element from an array.
- adding the element in between the array etc…

Requires complex logic to solve.

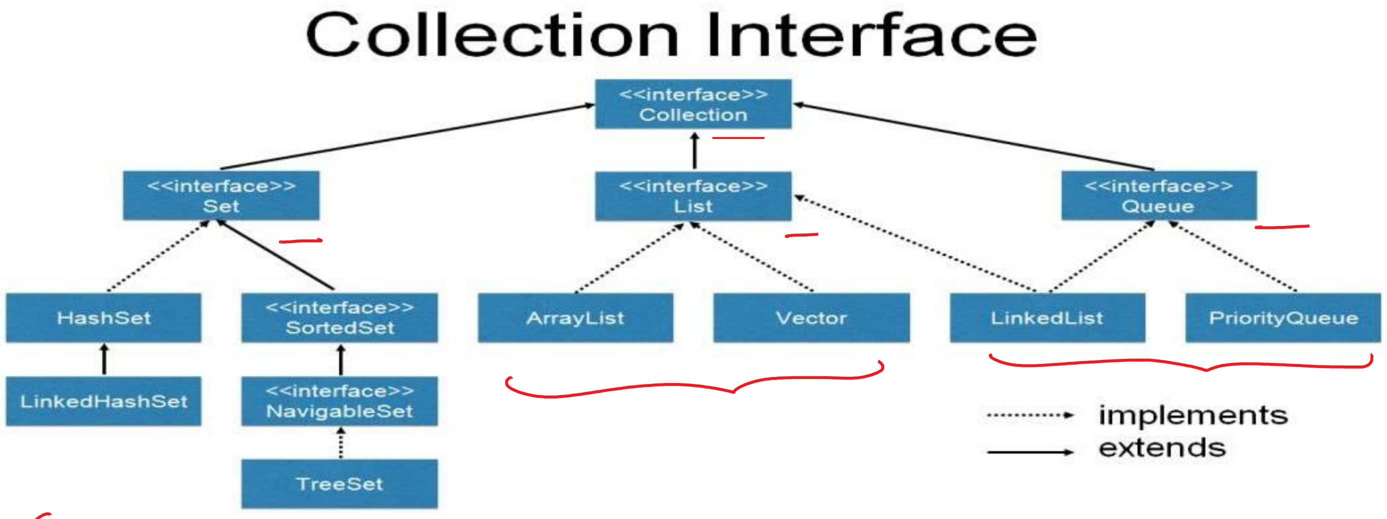
4) What is Collection?

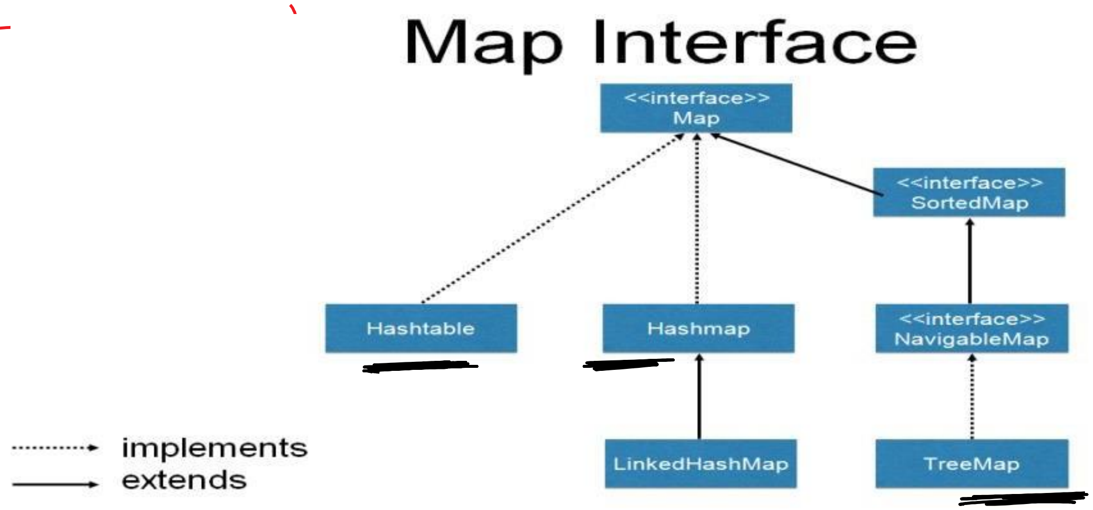
Collection Framework is set of classes and interfaces (hierarchies), which provides mechanism to store group of objects (elements) together.

It also provides mechanism to perform actions such as : (CRUD - operations)

- create and add an element
- access the elements
- remove / delete the elements

- search elements
- update elements
- sort the element





5) What is Array List?

- it is a Concrete implementing class of List interface.
- The characteristics of ArrayList is same as List.
- ArrayList is defined in java.util package.

Constructors:

ArrayList()	creates an empty ArrayList Object
ArrayList(Collection)	creates an ArrayList Object, and copies all the elements present in the given collection into the ArrayList created.

6) What are the characteristics of Array List?

1. Array List is a collection of objects.
2. Array List is an ordered collection of Objects. (It maintains the insertion order of elements)
3. Array List has indexing, therefore, we can insert or remove elements with the help of index.
4. Array List can have duplicate objects.

7) What is iterator?

iterator() is method which belongs to Iterable Interface.

iterator() method creates an Iterator type object and returns the reference.

iterator() method returns the reference in Iterator (interface) type

Iterator interface has 2 important methods to perform iteration

1. **hasNext() :** it checks whether there is an element to be accessed from the collection, if present it returns true, else it returns false.
2. **next() :** it is used to access the element, and moves the cursor to the next element. The return type of next() is Object.

8) What are the disadvantages of iterator?

We can iterate only once.

We cannot access the elements in reverse order.
We cannot remove an element from the collection, during iteration. We get an exception (ConcurrentModificationException).

9) What is ListIterator()?

listIterator() :

listIterator() is a method that belongs to List interface, listIterator method creates an object of ListIterator type.

return type of listIterator() is ListIterator interface.

ListIterator :

ListIterator is a sub interface (child) of Iterator interface, it is defined in java.util package.

10) What is set?

Set is an interface, it is a sub interface of Collection, therefore all the methods of Collection is inherited.

Set is defined in java.util package.

Characteristics Of Set :

- it is an unordered collection of elements. (the order of insertion is lost)
- Set does not allow duplicate elements.
- Set does not have indexing. Therefore, we cannot access, insert or remove based on index.
- We can access the elements of set only by using:

Using iterator()

Using for each loop

11) What is list?

List is an interface defined in java.util package.
List is a sub interface of Collection interface. Therefore, it has all the methods inherited from Collection interface.

Characteristics of list:

1. List is a collection of objects.
2. List is an ordered collection of Objects. (It maintains the insertion order of elements)
3. List has indexing, therefore, we can insert or remove elements with the help of index.
4. List can have duplicate objects.

12) What is string?

In java we store literals as objects.

String class is built in class defined in java.lang package.
String class is used to store string literals in java.

13)How do you create an object of string in java?

We can create an object for String class in 2 ways.

1) With the help of String literals
2) With the help of new operator

1) Object creating using String literals:

Anything enclosed within double quotes is known as String literal in java, everytime a String literal is created, an object of String class is created in the String constant pool area & its address is returned.

2) with the help of new operator:

we can create a string object with the help of ‘new’ followed by constructor of the string class.

The String object created using new will be stored inside heap area & not in the string pool area.

14) What are the characteristics of string class?

- String class is public.
- String class is final.
- String class extends to java.lang.Object
- String class implements.
  - a) Comparable interface
  - b) Serializable interface
  - c) CharSequence interface

15) How do you convert an array of character into string?

We convert a string into a character array by using 2 built in non-static methods of string class

1) toCharArray()
2) charAt(int index)

toCharArray()

it is a non-static method of string class which is used to convert a string into an array of characters.

Its return type is char[]

charAt(int index):

it is a non-static method of string class which is used to access the characters present inside the string. The charAt() method will return the character present in given index.

The return type of charAt() is char.

16) What is string builder?

whenever the string involves many manipulation in the application. It is recommended to store strings using either string buffer or string builder.

StringBuilder():

It creates an empty string builder object of capacity 16.

StringBuilder(int capacity):

It creates a StringBuilder object with the specified capacity

StringBuilder(String str):

It creates a StringBuilder object & initialize it with given String.

17) What are the disadvantages of string?

Since the string objects are immutable every time we try to do modification for a string a new object is going to get created inside String constant pool area. Therefore more memory as well as time is consumed, which reduces the performance of the application.

In order to avoid this whenever the string involves many manipulation in the application. It is recommended to store strings using either string buffer or string builder.