(1) What is 'java?

Ans:- Java is a high level programming language and it is a Multi-platform, Object-oriented programing language that can used as a platform independent.

(2) What is Variable? Types of Variables Explain with Example

Ans:- Variable is a Memory block which is used to Store the data.

Every Variable has 4 characteristics

1) Variable Name
2) Variable Type
3) Variable data
4) Variable Data type.

(3) Explain the difference blw Local Variable and Global Variable?

Ans:-

Local Variable	Global Variable
→ Local Variable is Only Created inside the Method	→ Global is created inside the class Outside the Method
→ Local variable initialization is Mandatory.	→ Global variable initialization isn't Mandatory

Local Variable Syntax

data type Variable Name = data;

Global Variable

~~public~~

Access Specifier Access Modifier datatype VariableName = data;

Global Variable can be divided into two ways:

1) static variable
2) Non-static variable.

(4) Can we print Local Variable without initialization?

Ans: No, we cannot print Local Variable without initialization.

(5) What is Keywords?

Ans:- Keywords are special words in java.
=> all the keywords are started with lower case.
Each and every keyword has Meaning.

(6) What are Identifiers? Explain Rules of Identifiers?

Ans:- Identifiers are the Names given by the developers.

Developer is Responsible for providing three identifier Names.

1) class Name
2) Variable Name
3) Method Name.

(7) What is the difference b/w while loop and Do-while loop?

Ans:- (1) while Loop Syntax:

```
while ( condition )
{
    ≡
    update
}
```

// First check the condition.
(if condition is true print the value and update the value)
// If condition is false exit from loop and update the statement.

(2) DO-while

```
Do
{
  // Body of the loop
  // statements execution
  updation
}while (condition);
```

First do the action and next check the condition after that print statement if condition true print and update. condition false only update.

(8) Explain why Java is platform Independent (or) Explain WORA Architecture?

Ans:- Java is a platform Independent why because once we write any java program we can anywhere any desktop. (like MacOs, Linux, Os etc.,)

- WORA Architecture (write once Run Anywhere).

(9) Explain why Java is not pure Object Oriented?

Ans:- Java is not a pure Object Oriented. Because Java cannot support primitive data types. Java is a fully Object Oriented programing language.

(Java is called ~~partially~~ pure Object Oriented when primitive data types are used wrapper classes)

(10) Explain the difference b/w = and == ?

Ans:- Double Equals to (==) is used to Compare the two values.

Single Equals to (=) is used to initialize the values of Variables in java.

(11) what is Object? what are properties of an Object

Ans:- An Object is also Member in java. It is also called as Instance. Every object will have State and behaviors.

State Represents ⟶ what Object knows?

Behaviors Represents ⟶ what Object does?

Object is created when we Create a Non-static Variable and Non-static Variable to initialize that Variable we Create Object.

(12) why should we Create Object-? what is the use?

Ans:- To initialize the Non-static variable and Non-static Methods that is why we Create Object.

(13) what is class?

Ans:- class is used to create of an Object. using Single class we can Create Multiple objects.

class is used for Creating blueprint for an object.

(14) Explain how class is used blueprint of Objec

Ans:-

(15) Explain difference blw Static and Non-static

Ans:- Static:- Static is a Access Modifier. Static Variable will have Single Copy Nature because we do not Create an Object of it also Static Method we do not Create Object for it. we can call directly.

Non-Static:-

A non-Static Variable (or) Method we Cannot access directly. If we want to access we must Create of an object and Every NSV will have Multiple Copy Nature.

(15) Explain when should we static and when should we use Non-static? Real-Example?

Ans:- When Should we use static:-

We can use static variable (or) static Method when there is only single copy Nature. And we can access directly. if we there is a single copy nature it can share the same copy to everyone but it cannot give Multiple copies to everyone. (Single copy of every object)

When Should we use Non-static:-

when there is a Multiple copies we can declare NSV and NSM. because NSV and NSM can have Multiple copy Nature. Every object will get different type of Objects we can create Non-static.

Example:- for static:-

India → Every Indian people they should mention their country India. So, we can create single copy static here.

Example for NS:-

Bank Account

AccNo → NSV

Acc Name → NSV

Acc Balance → NSV.

(17) Explain the difference b/w Stack Area & Heap Area?

Ans:- ~~Stack Area~~:-

When we run any java programe two Memory ways are created

Stack Area, Heap Area

StackArea:-

used For Execution purpose.

HeapArea: (used For Storage purpose)

JVM will make use of 3 Resources.

1) classLoader
2) MainMethod
3) Garbage Collector.

ClassLoader will load all the Static Members in Static pool Area.

MainMethod will execute the code statement by statement.

Garbage collector will clear all the heap area.

(18) Explain the Method? what is the Advantages of the Method & and its types?

Ans:- Method is a block of code (programme).
programme is a combination of data and a Operation.
all the operation will stored in a Method.

Advantages:-

Main reason of Method Code - reusability. we can write once and use it multiple times.

(19) Explain Method Overloading. explain. its advantages?

Ans:- Developing More than one Method with the same method Name but with different Parameters this process is called Method Overloading.

We can achieve Method Overloading by two ways:

(1) Should Maintaining the Same Method Name.
(2) Arguments should be differe from Method to Method.

Advantages

We go for Method Overloading "To perform Same Operation in multiple ways".

Method Overloading provides flexibility to the user to perform same Operation in Multiple ways.
User can choose the operation in his choice.

(20) Explain Method Overriding and Advantages of Method Overriding.

Ans:- Method Overriding:-

During inheritance process Subclass will get complete freedom to modify the method implimentation but it should be same method signature whatever Method Signature present in a Super class.

Method Overriding depends on two factors.

1) Is-A Relationship.
2) Subclass should Maintain Same Method Signature present in a Super class.

Advantages of Method Overriding:-

1) We can change a Method implementation of inherited Method without effecting other Subclass
2) Abstract
3) Interface
4) Polymorphism
5) Abstraction.

(21) Explain the diff b/w Method Overloading and Method Overriding.

Method Overloading	Method Overriding.
(1) Developing More than one method with the same method Name but with the different Arguments is called Method Overloading.	1) During Inheritance process Subclass will get complete freedom to modify Method implimentation.
(2) Method Overloading depends on two factors i) Maintaining Same Method Name ii) Argument list should be differe from Method to Method.	(2) Method overriding depends on two factors. 1) Is-A Relationship 2) Maintaing Same Method Signature present in Super class
(3) Method Overloading can be achieved in Same class.	(3) Method overriding can be achieved in different class by using Inheritance.
(4) We can Overload a Method Static (or) Non-static	(4) We can Override a Method only Non-static.
(5) It is compile time polymorphism	(5) It is Run-time polymorphism.

(22) What is the difference b/w class and Method?

Class	Method
1) A class is used to create an Object	1) Method is a combination of data & a Operation
2) Using Single class we can create multiple Objects.	2) Method describes the functionality of an Object.
3) class is Logical entity.	3) Method depends on the class.

(23) Explain Constructor and its types?

Ans:

Constructor:

Constructor is a Special Type of Method which gets executes only when we create an Object for it.

The Main purpose of constructor is initialize the NSV.

Syntax:

```
Access Specifiy ConstructorName (Args/No Args)
{
  ≡
}
```

There are two types of Constructors.

(1) Constructor with Arguement.

(2) Constructor without Arguement.

(23) Explain Default Constructor and User Define Constructor.

Ans: Default Constructor:-

A Constructor without emply body and without Args is called Default Constructor.

Incase developer doesn't Create any Constructor Compiler will Creates default Constructor.

ex:-

```
class Student
{
    public Student ( )
    {

    }
}
```

User Defined Constructor:-

A Constructor with Args and with body is called User defined Constructor.

(24) Explain Constructor Overloading?

Developing Multiple Constructors with different Arguement list is called Constructor Overloading.

we can achieve Constructor Overloading in within the same class and different class.

with same class, we can achieve.

with same class we can achieve by call to this statement.

with Multiple classes we can achieve by Super() (call to ~~this~~ super statement).

(25) Constructor chaining & its types Explain?

Ans:- A constructor calling another Constructor is called Constructor chaining.

we can achive Constructor Chaining within the same class by using call to this statement.

Syntax

this (Args/No Args)

we go for constructor chaining to avoid code - repeatation.

(26) Explain the difference b/w Method & Constructor.

Method	Constructor
1) Method can be executed when we call it.	1) Constructor can be executed when we create an object
(2) Method Name will not have same as class Name.	(2) Constructor Name will be same as class Name.
(3) Method should have Return type.	(3) Constructor should not have Return type.

(27) What is the difference b/w this and call to this Statement?

Ans:-

this.	this ( )
1) it is allowed only inside the constructor	1) it is used the Constructo as well as NSM
2) it is used to another constructor of the same class.	2) it is used to access the Non-static Members
3) it is used for code reusability.	3) it is used for code redability.
4) we can use call to this Statement only one time in Const.	4) we can use Construct Multiple times.

(28) Explain the diffe blw Super keyword and call to Super statement?

Ans:-

(29) Explain inheritance? and its types with Real time examples and its advantages?

Ans:- Inheritance:-

Super class will provide separate copy of its property, to each and every subclass is called inheritance.

Inheritance can be achieved in by using extends keyword.

There are 4 types of inheritance.

1) Single level inheritance.
2) Multi level inheritance.
3) Hierarchical inheritance.
4) Multiple inheritance.

Note:- Multiple Inheritance cannot be achieved by using classes. It cannot support in java.

Only Non-static Members will involve in Inheritance i.e, Only Non-static Members can be Inherited. because of its Multiple copy Nature.

Static Members cannot be Inherited why because Static Members will have Single copy Nature.

Super class static members are shared with all the Sub class.

Inheritance is also known as Is-A Relationship.

According to syntax we can Create an Object of Superclass. but from real world point of view, we should not create object of Superclass. because Super class object doesn't exist in the real world.

Only Subclass object should be created because Sub class object exists in the real world.

(1) Single level Inheritance :-

Having Single Super class and Single Sub class is called Single level Inheritance

```
class A
{
  public int i=10;
  public void display()
  {
    S.o.pln("i value is :"+i);
  }
}
class B extends A
{
  public int j=20;
  public void print()
  {
    S.o.pln("J value is :"+j);
  }
  S.o.pln("i value is :"+i);
}
class Main Method
{
  psvm(S[] args)
  {
    B b1 = new B();
    b1.print();
    b1.display();
  }
}
```

(2) Multi level Inheritance:-

Having Single Super class and Multiple Subclass is called Multi level Inheritance

```
class A
{
 public int i=10;
}
class B extends A
{
 public double j= 10.5;
}
class C extends B
{
 public int k = 100;
}
class D extend C
{
    public int L = 200;
   public void display()
    {
      S.o.pln("i value is:"+i);
      S.o.pln("J value is:"+j);
      S.o.pln("K value is:"+k);
      S.o.pln("L value is:"+L);
    }
class MainMethod
{  psvM(S[] args)
     {
       D d1=new d();
} } d1.display();
```

(3) Hierarchical Inheritance:-

Having Single Super class and Multiple Subclass is called Hierarchical Inheritance.

```
class A
{
  public void Run()
  {
    S.o.pln("Run daily");
  }
}
class B extends A
{
  public void Run()
  {
    S.o.pln("Run everyday");
  }
}
class C extends A
{
  public void Run()
  {
    S.o.pln("Run only evening");
  }
}
class MainMethod
{
  psvm(S[] args)
  {
    C c1 = new C();
    c1.Run();
  }
}
```

(4) Multiple Inheritance:-

It cannot be Supported in Java.
it can Support only Interface.

Advantages of Inheritance:-

1) Code Reusability
2) Abstract
3) Interface
4) Polymorphism
5) Abstraction.
6) Generalization.
7) Object casting

(30) Explain why Multiple Inheritance doesn't Support by using class in Java.

Ans: Java doesn't Support Multiple Inheritance in classes because of diamond problem.
Single Subclass cannot have Multiple Superclasses (Multiple Inheritance) Single Subclass doesn't extends Multiple Super class. there is a execution problem.

(31) Explain what is diamond problem in java.

Ans:- Single Subclass doesn't have Multiple Super classes. because of diamond problem.

Diamond problems are:-

1) Ambiguity in constructor chaining.

```
class A
{
  public A()
  {
    ≡
  }
}
```

```
class B
{
  public B()
  {
    ≡
  }
}
```

```
class C extends A, B
{ public C()
  {
    super()
  }
}
```

which Super class Constructor to call?

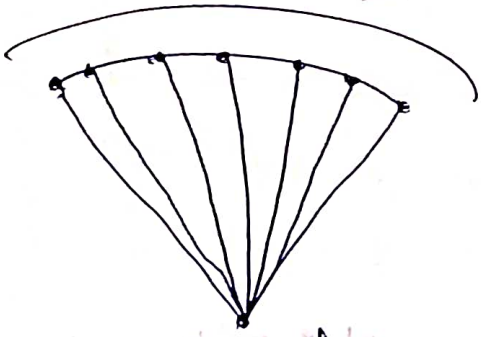
(2) Ambiguity in Execution problem?

```
class A
{
  public void move()
  {
    S.o.pln("move North");
  }
}
```

```
class B
{
  public void move()
  {
    S.o.pln("move South");
  }
}
```

```
class C extends A, B
{
  public void journey()
  {
    move();  → Error
  }
}
```

Super classes



Subclass.

which Superclass Method implimentation to execute?

(32) What is the use of extends?

Ans:- Extends keyword is used to inherite the multiple classes. (Super class to Multiple Subclass)

(33) Can we inherite Method? and Variables?

Ans:- A Subclass can inherits Variables and Methods from its Super class.

(34) Can we inherit Constructor in java?

Ans:- No, we cannot inherit constructor in java.

(35) What is upcasting and down casting? and its advantages with Real time examples?

Ans: Object Type casting:-

(1) Casting Means converting.

(2) Converting one Object Type to another. object type is called Object type casting.

Object type casting is classified into two ways:

(a) upCasting

(b) Down Casting.

Upcasting:-

(1) process of converting a Subclass Object type to Super class object is called upcasting.

(2) upcasting is possible because Subclass object will have properties of Super class.

(3) When Object is upcasted Sub class properties are ~~visible~~ hidden Super class properties are visible.

(4) Upcasting can be achieved in two ways

(a) implicit upcasting

(b) explicit upcasting.

Upcasting can be achieved in Implicit way because Subclass will have only one Super class hence, developer involvement is not Required. i.e., Subclass doesn't need any help of a developer to Convert to Super class.

Down Casting :-

process of converting Upcasted Object to Subclass Object is called "Down Casting".

For down casting we need to full fill two conditions.

i) Do not Create ~~two~~ new Object Rather Make use of existing object.

ii) Object should be already upcasted.

During down casting hidden Subclass properties becomes visible.

Down Casting can be achieved in two ways.

i) Explicit down Casting

ii) Implicit down casting is not possible.

down casting is explicit because Single Super class can have Multiple Sub classes. Hence developer involvement is neccessary and developer should help Super class to Convert to Sub class.

Example programe :-

```
class A
{
  public void Running()
  {
    So.pln("Run Daily");
  }
}
class B extends A
{
  public void Run()
  {
    So.pln("Run early Morning");
  }
}
class MainMethod
{
  psvm(s[] args)
  {
    A a1 = (A) new B(); // upcasting //
    a1.Running();
    B b1 = (B) a1; // Down Casting //
    b1.Run();
```

(36) What is Explicit Upcasting?

Ans:- Explicit upcasting is not possible because Subclass can have single Superclass and Upcasting can be done in Implicit way so Explicit Upcasting not possible.

(37) What is Implicit Upcasting?

Ans:- Sub class properties are hidden and Sub class can act like Super class then Sub class properties hidden Super class properties are visible.

(38) Why Down Casting is Explicit in Nature?

Ans:- Down casting is explicit because single Super class can have Multiple Sub classes. Hence developer requirement is necessary. Developer should help to convert Super class object type to Subclass Object type.

(39) Explain Generalization with Real tym example?

Ans:- The process of handle Method which can handle Multiple Objects of different types. These Methods are called Generalized Method and the process is called Generalization.

Steps to create Generalized Method.

1. Create static or NSM
2. Create Method with Arguement.
3. Arguement should be Superclass refurence Variable.
4. Call the Method and pass Subclass Object.

Note:-

Incase of Generalization, there will be always upcasting.

Advantage:-

In Generalization we go for Generalization to avoid code repeatation and to implement the code re-usability purpose.

```
class Company
{
    public void employee()
    {
        S.o.pln("Employee Details");
    }
    public void employee1()
    {
        S.o.pln("Employee1 Details");
    }
    public void employee2()
    {
        S.o.pln("Employee2 Details");
    }
}
```

```
class Developer extends Company
{

}
class Tester extends Company
{

}
class MainMethod
{
  public static void getEmpDetails(Company c)
  {
      c.employee();
      c.employee1();
      c.employee2();
  }
  public static void main(String[] args)
  {
      getEmpDetails(new Developer());
      getEmpDetails(new Tester());
  }
}
```

(40) Explain Specialization with Real time Example?

Ans:- Specialization:-

A Method which can handle Multiple objects of same type this process is called Specialization. this Methods are called Specialized Methods.

Steps to Create Specilized Methods:-

1. Create Static or Non-Static Methods
2. Create Method with Arguement.
3. Arguement should be reference Variable Type.
4. Call the Method and pass same class object as input.

Note:- we go for Specilization Methods to avoid Code repeatation and Impliment the code re-usability purpose.

Example:-

```
class BankAccount
{
    private long actNo;
    private int upiPin;
    private int MobileNO;
    long deposite=0;
    long withdraw=0;

    public BankAccount(long actNo, int upiPin,
                       int MobileNO)
    {
        this.actNo = actNo;
        this.upiPin = upiPin;
        this.MobileNO = MobileNO;
    }
    public void getBalance()
    {
        int balance = 10000;
        S.o.pln("your balance is:" + balance);
    }
    public void deposite()
    {
        S.o.pln("Entu your Acct No");
        S.o.pln("Entu your deposite Money");
        S.o.pln("Entu your upi PIN");
        int balance = balance + deposite;
        S.o.pln("your Amount Successfully
                 Deposited");
        S.o.pln("your balance ::" + balance);
    }
```

```
class MainMethod
{
    Public static void Customer(BankAccount ba)
    {
        ba.getBalance();
        ba.getBalance();
        ba.deposite();
    }
    psvm(String[] args)
    {
        Customer(new BankAccount());
        Customer(new BankAccount());
    }
}
```

(41) Explain Encapsulation and its Advantages with Real time Example?

Ans:- process of binding data Members with Member function is called Encapsulation.

(or)

Connecting data Members with member function into a single unit class this process is called Encapsulation.

Steps to achieve Encapsulation:-

(1) Declare Variables as private.

(2) Provide Access to the Variable through public Method.

(3) If program is not encapsulated, data Members will be misused by assigning invalid data.

(4) In Order to protect the data Members from invalid data we go for encapsulation.

Note:-

Challenging part in Encapsulation programme is to find the logic for Validation.

Programe for Encapsulation:-

```
class Calendar
{
  private int monthNumber;
  public void giveMonthNumber(int monthNum
  {
  this.monthNumber = monthNumber;
  }
  public void printMonthNumber()
  {
    if (monthNumber == 0)
      {
        S.o.pln("Try Again Invalid month
                 Number");
      }
    else
      {
        S.o.pln("monthNumber=:"+monthNumber
      }
}
class MainMethod
{
  psvm(s[] args)
  {
    Calendar c = new Calendar();
    c.giveMonthNumber();
    c.printMonthNumber();
  }
}
```

(41) What is the difference b/w Variable and reference Variable?

Ans:-

Variable	Reference Variable
1) It is a Memory block which is used to store the data.	1) it is created by developer by using call the methods and store the data from methods.

(42) What is final?

Ans:- Final is a keyword which is used to declare the Variables, Methods, classes declare as final.

final Means Constant (or) last Value.

Note:- If we declare Variables as final we cannot re-initialize for that Variables.

```
class A
{
    final public int i=10;
    psvm (s[] args)
    {
        public int i=20;   → Error. → Re initialization not possible.
        s.o.pln(i);
    }
}
```

(43) what happen if we declare class as final?

Ans:- If we declare class as final we cannot inherite that class.

(44) what happened if we declare Variable as final?

Ans:- If we declare Variable as final we cannot do re-initialization. etc not possible because Variable as final.

(45) what happen if we declare method as final?

Ans:- If we declare method as final we cannot do overridden for that method.

The Main Content is if we declare Method as final then that method should not be changed in that method by any outsidey.

(46) what is Method Binding?

Ans:- The process of connecting Method Signature with Method implimentation is called Method Binding.

There are 3 stages in java.

1) Coding
2) Compilation
3) Execution

(47) what is Compile time binding (or) Static binding (or) early binding?

Ans:- Compile time binding, Static binding, early binding these are all the same.

If methods are binded during Compilation stage this is called Compile time binding

Compile time binding is also known as Static binding because the connection blw Method Signature, Method implimentation cannot be changed.

Compile time binding is also known as Early binding because Methods are already binded before execution and Methods are ready execute.

(4e) what is Run-time Binding?

Ans:- If Methods are binded during execution stage this process it called Run-time binding.

Run time binding is also known as DYNAMIC binding. because connection blw Method signature and Method implementation can be changed based on the object creation.

Run time binding is also known as LATE binding because before binding the Methods JVM needs to perform other operation and then binding process will begin Since there is a delay for binding it's called Late binding.

(4a) Explain polymorphism and it with Real time example?

Ans:- Single Entity is having Multiple forms is called polymorphism.

polymorphism is classified in two ways

i) Run-time polymorphism

ii) Compile-time polymorphism.

Call to Overridden Method is decided during run-time based on object Creation this process is called Run-time polymorphism.

Run-time polymorphism is achieved by using
(a) Inheritance
(b) Method overriding
(c) Generalization
(d) Upcasting.

Compile-time polymorphism:-

Call to overloaded Method is decided during compile time based on the arguements this process is called Compile-time polymorphism Compile-time polymorphism can be achieved in two ways

(a) Static Method
(b) Method Overloading.

Advantages:-

→ By using polymorphism we can avoid code repeatation.

(50) Explain the diff b/w compile time and Run-time polymorphism?

Compile time Polymorphism	Run-time polymorphism
(1) Call to Overloaded Method is decided based on Arguement list this process is called polymorphism	(1) Call to Overridden Method is decided during Run-time based on Object Creation
(2) it can be achieved in two ways (i) static Method (ii) Method overloading	(2) it can be achieved in four ways (i) Inheritance (ii) Method overriding (iii) Generalization (iv) Upcasting.
(3) it can be achieved in Single class	(3) it can be achieved in Multiple classes
(4) it can be achieved in static complete Methods Abstract, interface not allowed	(4) it can be achieved in Complete Methods. it allows Abstract class, interfaces.

(51) What is abstract?

Ans:- Abstract Means Incomplete.

Abstract class can have only Method Signature without Method Implimentation, this process is Called Abstract. In complete Method is also known as Abstract Method.

Abstract is a keyword which reprasents Incompleteness.

(52) What is abstract class and abstract Method?

Ans:- Abstract class:-

Abstract class containg only abstract Methods. present in Super class.

we cannot create object for Super abstract class.

Abstract Methods:-

Abstract Methods cannot have Methods it can be used in only abstract class and it does not have body.

A Method can declared as abstract keyword within an abstract class and does not have a Method Implimentation.

when we need Method declaration in we can Impliment only Sub class.

(53) Is it possible to create object of abstract class?

Ans:- No, we cannot create object for abstract class.

But we can create an reference variable for abstract class.

Eg:-

```
abstract class Run
{
    public int i;
    public Run(int i)
    {
        this.i=i;
    }
}
class MainMethod
{
    psvm(s[] args)
    {
        Run r1;
    }
}
```

Run r1; reference variable for abstract class.

(57) What is Interface?

Ans:- Interface is the one of "TYPE DEFINITION BLOCK IN JAVA".

In Interface all the Variables are by default "final" and "static".

In Interface all the Methods are by "default public and abstract".

(58) What is Marker Interface?

Ans:- An empty Interface is called as Marker Interface.

Example:- interface Demo
{
}

(59) What is the difference b/w class and Interface

class	Interface
1) A class can be Inherited by using extends keyword.	1) Interface can only be implemented by using implements keywords.
2) A class can have any type of members like private, public, protected, default.	2) Interface can have only public members.
3) We can Create an Object for class.	3) Interface cannot be instantiated.

class	interface
it can allow SM, NSM	it can allow only SM
it can allow constructor	it cannot allow constructor

(60) Can we create constructor in interface

Ans:- No, we cannot create constructor in interface.

(61) which keyword we should use to inherite interface to interface?

Ans:- implements

(62) which keyword we should use to inherited class to interface.

Ans:- implements

(63) which keyword we should use to interface to class?

Ans:- Implements

(64) which keyword we should use to inherit class to class

Ans:- extends.

(65) Can we create object for interface?

Ans:- No, we can't create object for interface

Example:-

```
interface Sample
{
    void test();
}
class Demo implements Sample
{
    public void test()
    {
        s.o.pln("Test completed");
    }
}
class MainMethod
{
    psvm(s[] args)
    {
        new Demo();
    }
}
```

(66) what is abstraction ? with Real time example

Ans:- Hiding unnecessary details and provide only access to the necessary details this process is called abstraction.

(or)

Hiding Method implimentation details and providing access only to the Method Signature this process is called Abstraction.

(67) Steps to create Abstraction ?

Ans:- (1) Create an Interface
(2) Create a Method Signature
(3) Create an Implementation class.
(4) Implement the Method
(5) Create Helper class
(6) Create Helper Method (Static or Non-Static)
(7) Helper method will create an object of implimentation class and upcaste it to Interface and return the object.

(68) Write steps to use abstraction?

(1) Know the interface.

(2) Know the Methods of interface

(3) If is unecessary to know the implementation class.

(4) If is unecessary to know how the method implimented

(5) Know the Helper class.

(6) Know the Helper Method.

(7) Helper Method will give the Object.

(8) Recieve the object by using interface reference variable.

(9) Use interface reference variable to call interface Method.

(69) Is it possible to create Object for abstract class?

Ans:- No, we cannot Create only object for abstract class.

```
abstract class Run
{
    public int i;
    public Run(int i)
    {
        this.i = i;
    }
}
```

```
class MainMethod
{
    psvm(String[] args)
    {
        new Run(10);
    }
}
```

Error

(7o) What is the difference b/w abstract class and interface?

Ans:-

Abstract class	Interface
(1) abstract is a keyword	(1) interface is a keyword
(2) Subclasses extends abstract class	(2) Subclasses implements interface
(3) It cannot support Multiple inheritance	(3) It can support. A single interface can extends Multiple interfaces
(4) Abstract class can have final, static, non-static Variables	(4) Interface can by defaultly static and final variables
(5) Abstract class can contains constructor.	(5) Interface doesn't have constructor.
(6) Abstract class we can create only reference Variable, but we cannot create only object for abstract class.	(6) In interface we can Create only reference Variable and we cannot create only object for interface
(7) In abstract class we should do only extends Subclass.	(7) Here in interface we can do both extends as well as implements.
(8) It is a collection of abstract Methods and Concrete Methods	(8) It is a collection of only abstract method.

(70) What is the difference b/w Abstract class and Concrete class?

Abstract class	Concrete class