1) What do you know about JVM, JRE and JDK?

1) JVM: The JVM is responsible for interpreting the bytecode & translating it into machine code that can be executed by the underlying operating system. It also provides memory mangment security, and other runtime services necessary for executing Java applications.

2) JRE: JRE = rt.Jar + Java virtual machine.
> JRE is Platform dependent software that we can download from oracle site.
> To run Java application on clients machine we must install / deploy JRE.

3) JDK: Java Development Tools + Java Docs + rt.Jar + Java virtual machine
> Java Development tools are: JavaC, java, Javap, Javadoc, etc.
> Java Docs: HTML Pages, which contains help of core Java API.
> rt.Jar: It contains core Java API.
> JVM: It is abstract computer which manges execution of bytecode
> JDK is Platform dependt software that we can download from orcle site.
> Developer must install JDK to develop Java application.

Page No:
Date / /

2) is JRE platform dependant or independant
JVM, JRE and JDK all are platform dependant as it requires different configuration for different OS. However it is important to note that java is platform independant. The JIT (just in time) compiler converts Java byterode to machine code.

3) Which is ultimate base class in java class hirarchy ? List the name of methods of it ?
> object class is the parent class of all the classes in java.
> It is inside the package java.lang.
> so java.lang.object is package

> some important and useful methods of class object.
1) equals() :
It compares two objects and does the shallow comparision.
> signature : boolean equals (object obj)
> shallow comparision : It compare object references only and not the objects data
> Deep comparision : It compares objects data
2) Hashcode() : it is a hashkey associated with each object which is used for storing & retrival of objects through hashing technique as used

in map.
> Hashcode is not the address of object.
> Normally each object has uniq hascode but two object can have samcode.
> If objects are equal using equals() method then their hashcode() must also be equal.
> Hashcode can be negative value.

3) toString() : toString is a nonfinal method of Java.lang.object class.
> If we want to return state of object in string form then we should use toString method.
> If we dont define toString() method inside class then super class to string method will call. If any super class do not contain to string() method then object class to string method will call.

4) Explain narrowing and widening?

Assinging one data type into another data type we used Data conversion methods.

> first of all during conversion compiler sees whether the assignment or conversion is type compatible or not.

double = int → type compatible.
int = double → type compatible.
int = booleam → not type compatible

> Then its checks whether there is lossy conversion or not.

> large size data type = smaller size data (it is possible)

> Smaller size data type = larger size data type (possible lossy conversion)

> long = int ; → OK
int = long ; → possible lossy conversion
float = double ; → possible lossy conversion

> widening conversion (upcasting, implicit conversio) :

Assinging smaller data type into larger data type. It does not need typecasting.

larger data type = smaller data type
long = int.

Page No.
Date / /

Narrowing conversion : (downcasting explicit conversion)
Assinging larger data type into smaller data type. It needs typecasting.
Smaller data type = larger data type
int = long
> Typecasting can be used to do the assignment between different data type.

5) in System.out.println, Explain meaning of every coord ?
> System is a final class declared in java.lang. package.
> out is a refernce of java.io.PrintStream class. it is declared as public static final field inside system class.
> Println is a nonstatic method of Java.io.PrintStream class.
> Println Print output on console but it moves curesor to the next line.

6) Can you write java application without main function ? if yes how ?
> yes we can write Java application without main method by using a Static block. block in java is a groue of Statement that gets executed only once when the class is loaded into memory by Java Class loader. It is also known as static block initialization.

Date / /

7) What happen if we call main method in static block.

=> The static blocks always execute first before the main() method in java. because the compiler stores ~~him~~ them in memory at the time of class loading and before the object creation. Here the compiler executes all the static blocks first and after finishing the static block execution it invokes main() method.

8) How will you Print "Hello CDAC" statement on screen without semicolon.

> Every statem in Java must end with a semicolon as per the basics. However there are few scenarios when we can write a running program without semicolon.

> If we place the statement inside an if | for statement with a blank pair of parenthesis we dont have to end it with a semicolon.

> Using if else statements.

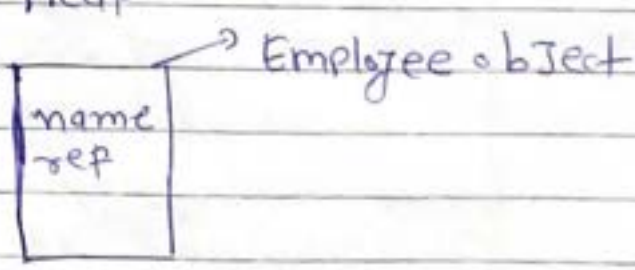
```
class Student {
    public static void main (String args[])
{
    if (System.out.println("Hello World")
                    == null) {
    }
    }
    }
```

9) How will you pass object to the function by reference.

In order to pass the reference we pass the object of class in the place of the actual parameter and the formal parameter of a class object type has the same refere to eachother thats why with the help of the formal parameter object of class any changes will be reflected in both objects formand and actual.

Declartion

```
Employee emp = new Employee();
```



Page No:
Date: / /

10) which are the rules to overload method in sub class ?

> The methods call is determined at compile time.

> Implements compile time polymorphism

> occuress between methods in same class.

> Have the same name but the Parameters are different.

Page No:
Date: / /

11) Explain contructor chaning ? And How can we achiere in java.

In contructor chain a contructor is called from another contructor in the same class this process is known as contructor overloading chaining.

It occures through inheritance.

when we create an instance of derived class all the contructor of inherited class (base class) are first invoked after that the contructor of calling class (derived class) is invoked.

> So we can achieve contructor overloading in two ways.

1) Within the same class we use this
2) from the base class : If we contructor belongs to different classes (parent and child) we use the super keyword to call the contructor from the base class.

> In this chaning the order of the constructor does not affect the output.

> by using the contructor chaining mechanism we can implement multiple task in a single contructor

> so whenever we face such type of problems we should use contructor chaining.

```
Public class student
{
  Student ()    // default contructor
  {

  }
student (string name, int rollno) //
                    parameterized Constructor
{
 this (name);
}
 Student (string name)
 {
   this ();
 }
public static void main (string args[])
 {  student s = new student ("John",
                              76);
}
}
```

Date / /

12) What is upcasting?
Process of converting reference of subclass into reference of saper class is called as upcasting

```
Employee emp = new Employee("RAM",
                39, 3778, 45000.50);
Person p = (Person) emp;
Person p = emp; // upcasting
```

> Super class reference can contain reference of subclass instance it is also called as upcasting.

```
> Person p = new Employee("RAM", 39,
                3778, 45000.50);
```

> If we want to minimize object / instance dependency in code then we should use upcasting.

13) Explain the diffence among throw & throws.

	Throw in java	Throws in Java.
1)	The throw Keyword is used to explicity throw an exception from within a block of code or a method	The throws Keyword is used in the method signature to declare the exception that a method can potentially throw.
2)	The throw Keyword can only throw a single exception at a time. As such it is possible to	The throw Keyword allows for the declartion of multiple exception that a function

	throw multiple exception simultaneously with throw	could throw
3)	The throw keyword is followed by the instance variable	The throws keyword is followed by the names of the exception classes.
4)	The throw keyword should be within body a method.	The throws keyword should be ued within the method signature.

14) Explain the diff between finilize & dispose ?

	dispose()	finalize.
1	it is defined in interface IDisposable interface.	It is defined in java.lang object class.
2	it is used to closeor release unmanged resources stored by an object like files or streams	it is used to clear up unmanged resources owned by the current object before it is destroyed.
3	It is declared as public	It is declared as private.
4	it is invoked by user.	it is invoked more slowly than dispose() method.

⑤ it is invoked very quickly	⑤ it is invoked more slowly than dispose method.
⑥ It executes immediate action and has no impact on performance of the site.	⑥ It has an impact on the performance of the site.

15) Explain diff between final, finally & finilize.

1) final : final is the keyword & access modifire which is used to apply restriction on a class method or variable
> finalkeyword is used with the class methods & variables.
> final method is executed only when we call it.

2) finally : finally is block in java exception Handling to execute the important code whether the exception occures or not.
> finally block is always related to try and catch block in exception handling.
> finally block is executed as soon as the try catch block is executed.

Page No.
Date / /

3) finilized: finilize is the method in java which is used to perform clean up processing just before object is garbage collected.

> finilize method is used with object.
> finilize method is executed must before the obj is destroyed.

16) in which case finally block doesnot execute?

1] Jvm crash or system.exit()
one of condition when finally block may not get executed is if the JVM crashes or the program is terminated using system.exit() method.

~~2) Infinite loop or other~~

```
Public static void main (String [] args) {
  try {
      System.exit(0);
  } finally {
      System.out.println("This will not
                          be executed");
  }
}
```

2) Infinite loop or other non terminating code
Another condition under which finally block not execute

```
public static void main (String[] args) {
    try {
        while (true) {
        }
    finally {
        System.out.println ("This will
            not executed");
    }
}
```

3) Thread killed:
In this finally block not execute when thread is killed using the stop() method

```
public static void main (String[] args)
throws InterruptedException {
  Thread thread = new Thread ()
try {
        Thread.sleep (1000);
  }
    finally {
    System.out.println ("This will
        not execute");
    }};

  Thread.start ();
    Thread.stop ();
}
```

Page No:
Date: / /

(4) System error:
finally block not executed when system fault occure like power outage or harware failure

```
Public static void main (String[] args)
  {
  try {
    } finally {
      System.out.println (" This will not
                          execute");
    }
    }
```

17) what do you know about final method.
when a method is declared as final it cannot be overridden by a subclass This is useful for methods that are part of class Public API and should not be modified by subclass

18) Why Java doesnot support multiple implementation inheritence
> because of the diamond problem in java is main reason Java doesnot support multiple inheritence in classes.

Page No.
Date / /

1) Explain dynamic method dispatch?

⇒ Dynamic method dispatch in java is a mechanism that helps to call an overridden at runtime by creating reference objects. It is the same as runtime polymorphism method dispatch in java uses upcasting. upcasting is a process to convert the ref variable of the child class reference variable of the parent class.

2) Explain marker Interface in java.

In Java marker interface also known as Tag interface are empty interface they do not have any variable or methods declared in them. marker interface in java acts as an indicator for an JVM to allow some special functions to the class which have implemented them.

1) clonable Interface

It is part of java.lang package. A clonable interface is used to make copy or clone of an object with different name. class must implement the cloneable interface in order to copy its object.

2) serialize interface:
serilization is a method that lets you read an object state out of memory and write to file or database

3) Remote interface:
A remote object is one that is stored on one computer but can be accessed from another computer. Java rmi package contain remote interface.

4) Custom marker interface:
A custom marker interface in java interface which has no methods or constants. it is used to provide additional information about class or object.

21) Explain fragile base class problem & how can we overcome it?
A franjile base class is common problem with inheritence which applied to java any other language which support inheritence.
> There are few methods of mitigating this but no straight forward method
> so make label all classes as final

unless you are specifying intending to inherit from them. for those to intend to inherit from design them as if you were designing API.

> The fragile base class problem has been blamed on open recursion (dynamic dispatch of methods on this) with the suggestion that invoking methods on this default to closed recursion (static dispatch, early binding) rather than open recursion (dynamic dispatch, late binding) only using open recursion when it is.

Page No.:
Date. / /

22) Diff between checked and unchecked Exception in java.

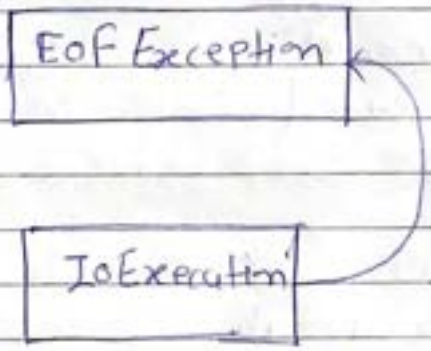
	checked exception	unchecked exception
①	checked exception happend at compile tim when the sources code is transformed into an executable code.	unchecked exception happen at runtime when the executable program starts running.
②	The checked exception is checked by the compiler	These types of exception are not checked by the compiler.
③	checked exception can be created manually	They can also be created manually.
④	This exception is counted as a sub class of the class	This exception happens in runtime and hense it is not included in the exception class.
⑤	Java virtual machine requires the exception to be caught or handled	Java virtual machine does not need the exception to be caught or handled

Page No.
Date / /

23) explain exception chaining.

chained exception occurs when exception causes another exception. The original exception is the cause of the second exception.

> In java a chained exception is an exception that is caused by another exception. chained exception are associted such that the previous exception causes each exception in the chain. It can help debug as it can help us track down the root cause of an error.



Exception chaining.

>We can use exception chaining in situation where another exception cause one exception. However, it is important to note that chaining can make our code more difficult to read and understand. therefore we should use exception chaining sparingly and only when necessary.

Page No.:
Date: / /

(25) which are the references type in java.

In java these are ~~four~~ types of References

1) In java there are are four types of References diffientited on way by which they are garabe collected.

1) Strong Reference:
This is default type / class of Ref object. Any object which has an active strong Reference are not eligible for garbage collection.

2) weak Refernce: weak Reference objects are not the default type / class of Refence object and they should be explicitly specified while using them.

3) phantom References: The objects which are beign referenced by Phantom refemeces are eligible for garbage collection.

4) Soft References: In soft refernces even if the object is bee for garbage collection then also it is not garbage collected