Q1) What do you know about JVM, JRE and JDK.?

⇒ ① JDK is the java development kit which includes development tools, Documentation, support or required libraries & execution environment. These are required for any program to run.

② JDK itself contains JRE & JVM. JDK is platform dependent as it is different for Linux & windows. In order for development JDK is must.

③ JRE is Java Runtime enviroment which is responsible for the execution of code. From Client's point of view if client just want to run the application on his device/ machine then JRE we must installed on their machine. JRE is platform independent means .class generated will be run on any machine.

④ JVM is java virtual machine. JVM is responsible for interpreting the bytecode to machine code. JVM is also known as mini operating system. as it provides memory management, CPU scheduling, etc.

② Is JRE platform dependent or Independant.

⇒① JRE (Java Runtime environment) is independant platform. It comes with JDK but we can install it seperately.

② JRE is responsible for the execution of code. The .class generated by JVM is a file which can be run on any enviroment irrespective of OS. That is why JRE is platform independent.

③ Client just have to install JRE & java application can run on it.

Q3) which is ultimate base class in java class hierarchy 2. List the name of methods of it 2.

⇒① If we don't extend any class in our class

eg:-

```
class Demo{
}
class program{
    Public static void main (string arg[]){
     // TODO
    }
}
```

In above code if we don't extend anything in our class then java compiler will by default extends Object of java implicity.

```
class Demo extends Object{
}
```

(2) Object class is superclass of ~~the~~ all the classes in java. Object class is concrete class present in java.lang package.

(3) object class does not extend any other class or implement interface. Hence it is also called as ultimate superclass.

(4) Object class have overall 11 methods. or main 5 methods.

(i) toString - return string in form of hashcode.

(ii) equals - compare the addresses.

(iii) hashcode - gives hashcode

(iv) Clone - creates copy of object

(v) finalize - use to deallocate the memory.

Q4) which are reference types in java?

⇒ reference types are used store ~~data~~ references i.e memory address to the instance.

(1) class types :- These are referenced to the instance when we create instance of class with new keyword.

```
Demo D = new Demo();
```

This creates memory for Demo instance on heap & that memory address is being stored in 'D'. 'D' is reference.

Also there are Array types & enum types

Q5) Explain Narrowing & widening?.

⇒ ① Narrowing :- The process of converting higher data type to lower data type is called Narrowing.

In case of narrowing we need to type cast explicitly. otherwise it will give "lossy convertion error".

eg:- double d = 20.5;
int i = d; // Narrowing.

② Widening :- The process of conver lower type into higher type. is called as widening.

In case of widening compiler will type cast automatically.

eg:- byte b = 10;
int i = b; // widening.

Q6) How will you pring "Hello CDAC" statement on screen, without semicolon?

⇒ We can achieve this by wing if statement, simply we will print system.out in if itself

eg:- if (System.out.println("Hello CDAC") == NULL)

In this way we can do it.

Q7) Can you write java application without main function ?

⇒ No, it is not possible to write java application without main function.

In java main function is entry point function. While execution the class with main & public keyword class & it will compile without main method but at the runtime .class file will load & search for the main method in the class which is public. & if it does not find main method then it will give "main class not found" error.

Q8) What will happen if we call main method in static block ?

⇒ It will show compiler error as static blocks are use to initialize static field and calling any static method of class. When we write main method in it it will give error of illegal expression. Static block is called only once after loading of .Class file.

Q9) In System.out.println, explain meaning of each & every word. 2

⇒ ① System is a final class in java.lang.package.

② out is a reference of printstream class. It is public static final field

③ printstream is class in java.io Package.

④ & println is non-static method of java.io.printstream class

⑤ This is use to print on console.

Q10) How will you pass object to function by reference.?

⇒ we can't pass object to function by reference like this.

```
Demo d = new Demo();

Demo.fun(d);
```

Here we are passing reference of class Demo as argument.

```
static void fun(Demo d)
{
//todo
}
```

we can access & modify the value of Demo by using this reference.

Q11) Explain constructor chaining & In java.

⇒ ① constructor is a block which get implicitly called by compiler when we create an instance. It is basically use to initialise the non-static member of class.

② There can be more than one constructor in class. parameterless & parameterized and calling one constructor from another constructor is known as constructor chaining.

③ This can be achieved by this(), by writting this() or this(int a, int b) we can call any constructor and this() should be on very first line otherwise it won't work.

```
class Demo{
    Demo(){

    }
    Demo(int a, int b){
      this();
    }
    Demo(int a, int b, int c){
      this(10, 20);
    }
```

Q12) which are the rules to overload method in sub class ?

⇒ (1) In case of method overloading the main point is rule is that the name of method should be same.

(2) After that the number of arguments should be different. or the type of arguments should be other type.

(3) In method overloading return type doesn't matter as there is no compulsion to return or catch the data.

eg:-

```
int sum(int a, int b){}
Void sum(int a, int b, int c){}
int sum(float a, int b){}
```

Q13) Explain difference between finalize & dispose.

⇒ finalise is use to deallocate the resource & dispose too. But main difference is that finalize is called by garbage collector when instance is no longer in use.
and on the other hand Dispose method can call implicity to release resources.

Q14) Explain difference between final, finally & finalize.

⇒ ① Final :- (i) Final is use to make a field constant in other way if we don't want to change value of field then it should made final.

(ii) Final class is also there which we can't inherit and can be make only instance of.

iii) final method is a method which can't be override.

② Finally : It is block with try & catch this block will be executed even if an exception is thrown.

③ finalize :- It is method of Object class which is use to deallocate the resources & this method is implicitly called by garbage collector.
We can explicitly call it.

Q1.

Q)15) Difference between checked exception & unchecked exception?

⇒ Checked exception simply happen at compile time while compilation. & runtime exception is unchecked exception

eg:- int a = 10

"semicolon is missing" :- checked at compile time

```
int a= 10, b= 0;
result = a/b;
```

eg: sop(result);

"arithmetic exception" :- unchecked runtime exception.

Q16) Explain exception chaining?

⇒ when one exception causes another exception it is exception chaining.

Q17) Explain difference between throw & throws

⇒ ① If there is a method which can lead to an exception then throws keyword is used & If we have to throw specific exception logically throw is used.

② throw can throw only one exception at a time but throws can used to declare multiple exception.

Q18) In which case, finally block doesn't execute?

⇒ finally is a block which executes even there is exception or not. But there is one case in which it is not executes i.e If the code is terminated inside the try or catch block only then finally block won't be executed.

Q19) Explain upcasting?

⇒ Consider following

```
class A {
    int a;
}
class B extends A {
    int b;
}
```

In main ∴

```
A a = new B(); // upcasting
```

If a class ~~of~~ parent is refering to child class then it is upcasting.

It is use to access parent class member easily but child class ~~prace~~ methods will not completely accessible.

Q20) Explain Dynamic method dispatch?

⇒ The process by which a call to an overriden method is resolved at runtime is Dynamic method dispatch.

simply if method of parent class is overrided by child class then while execution which method to be called is resolved at runtime.

Upcasting plays major role in this.

Q21) What do you know about final method.

⇒ final method is a method which can't be overide by subclass. If we don't want the subclass to override the method then we should declare it final.

Q22) Explain fragile base class problem & how can we overcome it?

⇒ The changes made in parent class unintentionally break or disrupt the functionality of child class. This is fragile base class problem.

Make the instance variable private & to use access modifiers to parent class. is the one of the solution. to avoid this