Java Interview Questions

Date
Page 1

1) What is JAVA ?

⇒ Java is a high level, object oriented, robust, secure, platform independent, high performance and portable programming language. It was developed by ~~Jason~~ James Gosling in 1991. It is also known as platform as it provides its own JRE and API.

2) Features of JAVA :-

i) Simple :- Java is easy to learn. The syntax of java is based on c++ which makes it easier to write program in it.

ii) Object-oriented :- Java follows object-oriented paradigm which allows us to maintain our code as the combination of different types of objects.

iii) Portable :- Java is write once and run anywhere. We can execute java program on every machine. Java program (.java) converted to bytecode (.class) which can easily run on every machine.

iv) Platform-independent :- Java comes with its platform on which its code is executed. Java doesn't depend upon the operating system to be executed.

v> Secured :- Java is secured because it doesn't use explicit pointers. Java also provides concept of ByteCode and Exception Handling which makes it more secured.

vi> Robust :- Java is strong programming language as its uses strong memory mgt. The concept like automatic garbage collection, exception handling make it more robust.

vii> Architecture Neutral :- Java is architecture neutral as it is not dependent on architecture In C, size of data types may vary according to architecture (32 bit or 64 bit)

viii> Interpreted :- Java uses JIT interpre-ter along with the compiler for program execution.

ix> High Performance :- Java is faster than other traditional languages because java bytecode is "close" to native code.

x> Multi-threaded :- The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares common memory area.

xi) Distributed :- RMI and EJB are used for creating distributed applications.

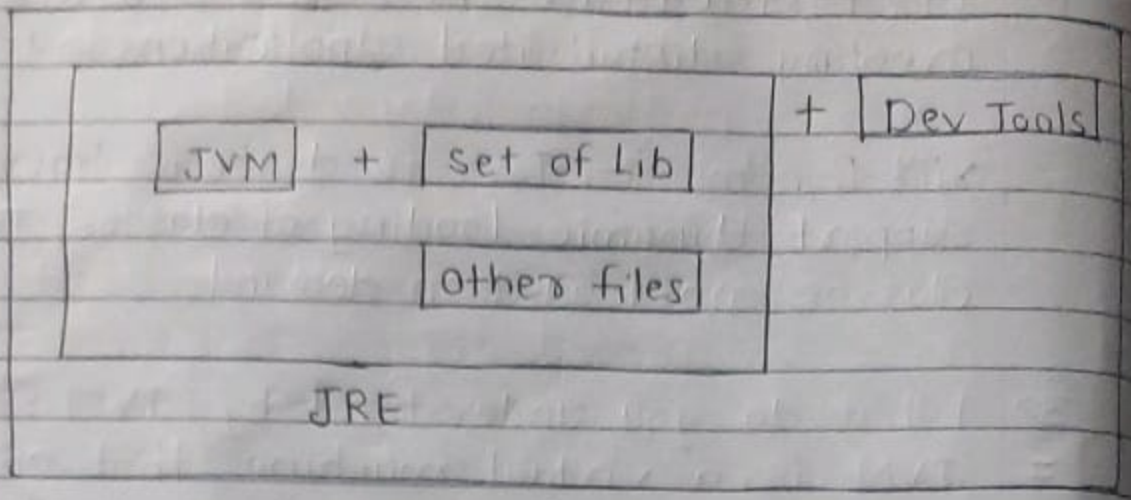
xii) Dynamic :- Java is dynamic language. It support dynamic loading of classes. It means classes are loaded on demand

3) What do you understand by JVM?
= JVM is a virtual machine that enables the computer to run java program. JVM acts like a run-time engine which calls the main method present in Java code. The java code is compile by JVM to be a bytecode which is close to native code.

4) How many types of memory allocated by JVM?
⇒ 1) Heap - whenever object is created
2) class Area
3) Static
4) Native method Stack
5) Program counter register.

5) What is platform?
⇒ Platform is a set of hardware and software environment in which piece of SW is executed. Java is software based platform.

6> JDK / JRE / JVM



	JDK	JRE	JVM
1>	Java Development Kit	Java Runtime Environment	Java Virtual Machine
2>	Platform dependent	Platform dependent	Highly platform dependent
3>	It is superset of JRE	It is a subset of JDK	It is subset of JRE
4>	JDK comes with installer	JRE only contain environment to execute source code	JVM bundled in both JDK and JRE
5>	It contains dev tools, debug tools and other tools	It contains set of libraries and other supporting JAR to execute Java code.	It does not contain any tools.

7>

8>

7) JIT Compiler ?

At Compile Time

Source Code .java → compiler → Byte Code

At Runtime

Native Machine Code ← JIT compiler

The java runtime environment (JRE) is responsible for executing source code and it contains JIT compiler that does the performance optimization It converts the Byte Code into Native machine code at runtime.

8) What gives Java its 'Write Once Run Anywhere' nature ?

⇒ The bytecode. - Java compiler converts java source code file (.class) into byte code which is intermidiate language between source code machine code. The bytecode is not platform specific and can be executed on any computer.

9> Is Empty .java file name is valid ?

⇒ Yes.

```
class A {
    ---
}
```

To compile :- javac .java
To run :- java A

10> What is the default value of local variables?

⇒ Local variables are not initialized to any value, not primitive not object references

11> What are the different access specifiers in Java ?

⇒ public, private, protected, default.

- public → within class, package, outside package, outside package by subclass.
- private → within class only
- protected → within class, package, outside package by subclass only
- default → within class & package.

12) What is the purpose of static methods and variables in java?

⇒ The methods or variables that are defined as static are shared among all the objects of the class. The static is a part of class and not the objects. The static variables are defined in class area and we don't need to create object of class to access such variables.

ex:- college name of students.

13) What are the advantages of packages in Java?

⇒ i) Packages avoid name clashes
ii) Packages provides easier access control.
iii) We can have classes access only within the package.

14) What is the initial value of object references which is defined as an instance variable?

⇒ All object references are initialized to null.

15) What is constractor?

⇒ Constructor is a special type of method that has same name as of class name and is used to initialize state of object. Every time object is created using new keyword, defautt constructor is called. It must not return any explicit value.

16) How many types of constructor are used in java?

⇒ Default constructor, Parameterized constructor

17) What is the purpose of default constructor?

⇒

```
class Student {
    int id;
    String name;
    Student_Info() {
        S.o.p.(id + " " + name);
    }

    p.s.v. main(String args[]){

    Student s1 = new Student();
    Student s2 = new Student();
    s1. Student_Info();
    s2. student_Info();
    }
        }
```

output :- 0 null
0 null

18) Does constructor return any value?
⇒ instance of a class

19) Can constructor inherited?
⇒ No

20) Can you make constructor final?
⇒ No.

21) Can constructor overloaded?
⇒ Yes, by changing no. of arguments or by changing data types of arguments.

```
class Test {
    int i;
    public Test(int k) {
        i = k;
    }

    public Test(int k, int m){
        i = k+1;
        m = k+2;
    }

public class main {
    p.s.v.main(String args[]) {
    Test t1 = new Test(10);
    Test t2 = new Test(10,20);
    S.O.p.(t1.i); S.O.p(t2.i);
}
}
```

22>	Constructor	Method.
i>	Constructor must have same name as of class name	i> Method name may or may not be the same.
ii>	It must not have any explicit return type	ii> It should/ must have a return type
iii>	Constructor cannot be overrided	iii> Method can be overrided.
iv>	Constructor is used to initialize State of object.	iv> Method exposes the behaviour of the object.
v>	Constructor invoked automatically when the objects created using new keyword.	v> Method needs to be exposed explicity
vi>	Java compiler provide default constructor if you dont have any constructor in class	vi> Compiler does not provide any default method.

23) Static Variable and Static Method.

⇒ Static variable used to refer to common property of all objects
- Static method belongs to class rather than object
- It can be called using class name.
- A static method can access and change value of static variable.

24) What are the restrictions applied to java Static methods?

⇒ i) Static method cannot use not static variable or call non-static method directly
ii) this and super keyword cannot used in static context as they are non-static.

25) Why the main method is static?

⇒ Because objects are not required to call the static method. If we make java main method non-static, JVM will have to create object first and then call main() method which will lead to extra memory allocation.

26) Can we override static method?

⇒ No.

27) Can we overload static method ?

⇒ Yes

```
class Test0{

    void display(int a){
        S.O.p.(a);
    }
}

class Test1{
    void display(int a, int b){
        S.O.p.(a + " " + b);
    } p.s.v.main(String args[]){
}       Test0.display(10); Test1.display(10,20);
```

28) Can we execute java program without main() method ?

⇒ It is possible before JDK 1.7
Since JDK 1.7 it is not possible.

29) Can we make constructor static ?

⇒ Constructor is invoked when object is created. If we try to make constructor static, it will throw compile-time error

30) Can we make abstract methods static?
⇒ In java, if we declare abstract methods as static then it will become part of class and we can directly call it which is unnecessary. Calling undefined method completely useless therefore it is not allowed.

31) Can we declare static methods and variables inside abstract class?
⇒ Yes

32) What is this keyword in JAVA?
⇒ 'this' keyword is a reference variable that refers to current object.
i) this can be used to refer current class instance variable.
ii) this can be used to invoke current class method.
iii) this can be used to invoke current class constructor
iv) this can be used to return current class instance from the method.
v) It can be used as argument in method call
vi) It can be passed as argument in constructor call.

33) Can 'this' is used to refer static members?
⇒ Yes, but it not good practice

34> What is inheritance ?

⇒ Inheritance is the mechanism by which one object aquires all the properties and behaviour of another object of another class It is use for code reusability and method overriding.
1> Single 2> Multi-level 3> Multiple
4> Hybrid 5> Hierarchical.

Advantages :-

1> Inheritance provides code reusability.
2> To achieve runtime polymorphism.
3> Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
4> Code is less and more readable.

35> Which class is superclass of all classes ?
⇒ Object class.

36> Why pointers are not used in Java ?
⇒ Because pointers are unsafe and complex to understand.

36) Super keyword

⇒ 'Super' keyword in java is a reference variable that is used for referring immediate parent class object.

Whenever you create instance of a subclass, instance of parent class also created which is referred by super reference variable.

```
class Animal {
    Animal() {
        S.o.p.("Animal is created");
    }
}
class Dog {
extends Animal
    Dog() {
        S.o.p.("Dog is created");
    }
}
class Test {
    p.s.v. main(String[] args) {
        Dog d = new Dog();
    }
}
```

⇒ Animal is created
Dog is created.

- Super can be used to refer to immediate parent class instance variable.
- Super can be used to invoke immediate parent class method.
- Super can be used to invoke immediate parent class constructor.

37) Can we use this and super in a constructor?
⇒ No.
this must be first statement in a constructor

38) Can we override the overloaded method?
⇒ Yes.

39) Can we override private methods?
⇒ No, because scope of private method is limited to the class and we cannot access them outside the class

40) Can we change the scope of overidden method in subclass?
⇒ Yes, we can change the scope of overridden method in the subclass. But we cannot decrease the readibility. —
- private - private can be change to public, protected and default.
- default - default can be changed to public
- protected - can be changed to public and default
- public - public will remain public.

41) Can we modify 'throws' clause of superclass method while overriding it in subclass?

⇒ Yes, but -

1) If superclass doesn't declare exception, subclass overridden method cannot declare checked exception, but can only declare unchecked exception.

2) If superclass declare exception, then subclass cannot declare parent exception of exception declared by superclass.

42) Final variable / class / method.

⇒ Stop value change
Stop method overriding
Stop inheritance

43) Final blank variable

⇒ A variable declared as final and is not initialized.

44) Can you declare main method as final

⇒ Yes, p.s.v.m final void main(String args[])

45) Can we override main method

⇒ No, because main method is also a static method.

46) Can we declare constructor as final?
⇒ The constructor can never be declared as final, because it is never inherited. If you try to do so, it will throw compile time error.

47) Can we declare interface as final?
⇒ No, you cannot declare interface as final because interface must be implemented by some class to provide its definition. Therefore, there is no sense to make it final. However, if you try to do so it will throw compile time error.

48) Difference between final and abstract method
⇒ Abstract method cannot be final as we need to override them in subclass to give its definition.

49) What is Java instanceOf operator?
⇒ The instanceOf in java also known as type comparison operator because it compare instance with type.

```
class Simple {
    p.s.v.main(String args[]) {
        Simple s = new Simple();
        S.o.p(s instanceOf Simple);
}
}
```

51) Can we achieve runtime polymorphism by data members?

⇒ No, we can override data functions, not data members.

52) What is abstraction?

⇒ Abstraction is the process of hiding the implementation details and showing only functionality to the user.
Abstraction enable you to focus on what object does instead of how it does.
• Interface • Abstract class.

53) What is the difference betⁿ abstraction & encapsulation?

⇒ Abstraction hides the implementation details, Encapsulation wraps the code and data into a single unit.

54) What is abstract class?

⇒ A class that is declared as abstract is known as abstract class. It needs to be extended and its method needs to be implemented.
It cannot be instantiated.
It can have abstract, non-abstract, static methods and constructors.
It can also have final methods which will force subclass not to change body of method.

55) Can there be an abstract method without an abstract class?

⇒ No, if there is a abstract method in a class, the class must be abstract

56) Can you use abstract and final both with a method?

⇒ No, because we need to override the abstract method to provide its implementation, whereas we can't override the final.

57) Can you declare interface method static?

⇒ No, because methods of interface are abstract by default and we cannot use static & abstract together.

58) Can interface be final?

⇒ No, because interface needs to be implemented by other class and if it is final, it can't be implemented by any class.

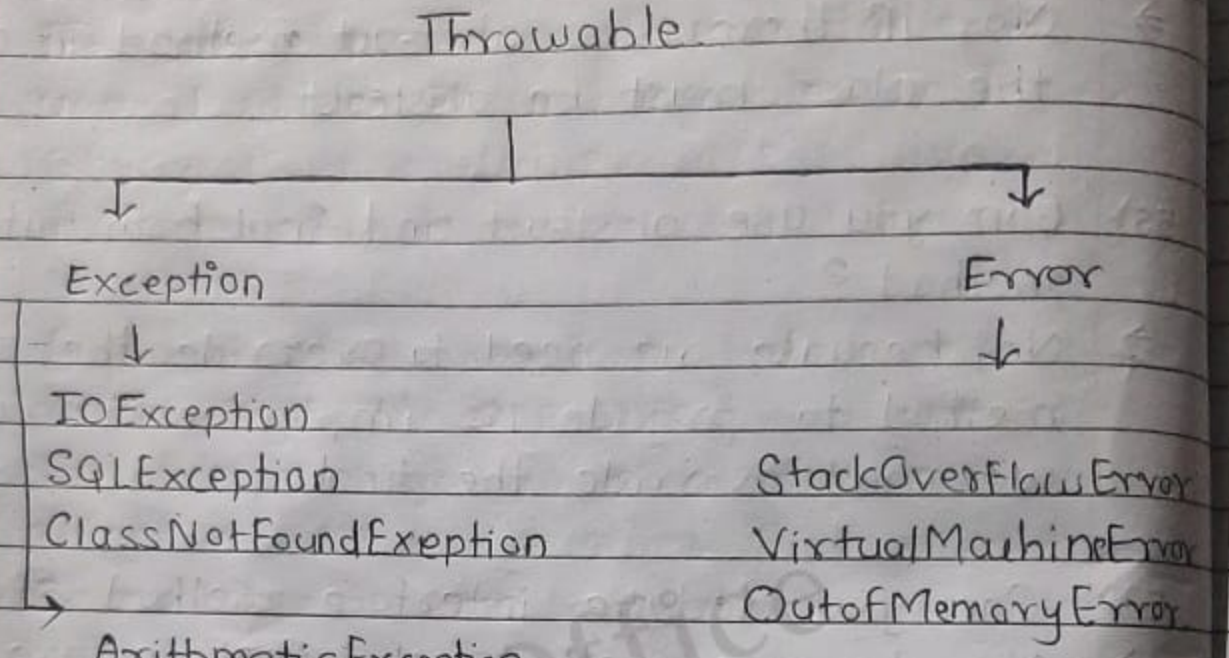
59) How many types of exception can occur in program?

⇒ i) checked - exception - compile time
SQLException, ClassNotFoundException

ii) Unchecked Exception - Runtime
ArrayIndexOutOfBounds, NullPointerException

iii) Error
- Error cause the program to exit since they are not recoverable - OutofMemoryError, StackOverflowError

61) Hierarchy of Java Exception Classes



- Unchecked Exception
  The classes that extends Runtime exception are known as unchecked exception, they are not checked at compile-time

- Checked Exception
  The classes that extends Throwable classes except RuntimeException and Error are known as checked exception.

62) What are base class of Error and Exception?
⇒ Throwable

63) Is it necessary that each try block must be followed by catch block?
⇒ It should be either follow by catch block or by finally block.

64) What is Finally block?
⇒ The 'finally' block is used to execute important code of program. It is executed whether an exception is handled or not.
- Finally block is mainly used to place the clean-up code such as closing file or closing database connection.
- For any try block, there can be zero catch block, multiple catch block, but only one finally block.
- Finally block will not be executed if program exit by calling System.exit(), by causing fatal error.

65) What is String pool?
⇒ String pool is the space reserved in heap memory that can be used to store strings. The main advantage of using string pool is whenever we create string literal, JVM checks the string pool constant first, if string already exist, it return the reference of that string. If string doesn't exist, it return new string reference. It saves memory, avoid duplicacy.

66) What is the meaning of immutable regarding String ?

⇒ The simple meaning of immutable is unmodifiable or unchangable. In Java, String is immutable, once string object is created, its value cannot be changed.

Suppose there are five reference variables all refer to the one object, if one reference variable change value of object, it will affect all ref. variable. That is why String objects are immutable in java.

67) How many ways we can create String obj?

⇒ 1) String Literal

```
String s1 = "welcome";
```

2) new keyword

```
String s2 = new String("welcome");
```

68) How many objects created ?

```
String s1 = "India";
String s2 = "India";
String s3 = "India";
```

⇒ only one

== checks references
.equals checks contents.

Date
Page 25

69) How many objects are created ?
⇒ String S1 = new String("India");

⇒ Two - one in string constant pool and other in non-pool (heap).

70) How can we create immutable class in JAVA ?
⇒ By making class and all of its member as final.

71) Why charArray is preferred over String to store password ?
⇒ String stays in the string pool until the garbage is collected. If we store the password into the string it stays in memory over long period of time and anyone having access to memory dump can extract the password as clear text.

On the other hand, using char array allows us to set it to blank whenever we are done with the password. It avoids the security threat and enables us to control memory.

72) What is Garbage collection ?
⇒ Garbage collection is the process of reclaiming the unused runtime objects. It is performed for memory management. In other words, we can say that it is a process of removing unused objects from the memory to free up space and make this space available for JVM.

73) What is gc() method?

⇒ The gc() method is used to invoke garbage collector. This method is found in System & Runtime classes.

```
System.gc();
```

74) How object is unreferenced?

⇒ ① By nulling

```
S1 = null;
```

② By assigning reference to another.

```
S1 = S2;
```

③ By anonymous object

```
new A();
```

75) Is Java pure object oriented?

⇒ Java uses primitive data type and hence is not pure object oriented language.

76) Can class be declared as protected?

⇒ No, only methods can be declared as protected.

77) Can a source file contain more than one class declaration?

⇒ Yes, source file contain any no. of class declaration but only one class can be declared as public.