1. What is Java? / Explain some features?

Ans : Java is a high-level, compiled, strongly typed object-oriended programming (OOP) language.

Features of Java:

* Java is Platform Independent.
* Write once, run anywhere
* Java comes up with garbage collected by default
* Java has the 2nd fastest compiler embedded in it
* it is the purest form of Object Oriented Programming.

1. What is JRE / JDK / JVM?

**JVM (Java Virtual Machine):**During the development of Java, the developers faced many problems because as the OS gets updated, the system was not able to run the previously written codes. To rectify this problem, JVM came into existence. It is an abstract machine that is present in the user’s computer and converts the **bytecode into machine code.**

**JRE (Java Runtime Environment):**JRE refers to a software package in which java bytecode can be executed. It implements the JVM (Java Virtual Machine) and provides all the class libraries and other support files that JVM uses at runtime.

**JDK (Java Development Kit):**This tool is necessary to compile, document and package Java programs. The JDK includes JRE which contains tools for Java programmers. Along with JRE, it includes an interpreter/loader.

**JSE :** Java Platform, Standard Edition ([Java SE](https://www.oracle.com/java/technologies/java-se.html)) lets you develop and deploy Java applications on [desktops](https://www.oracle.com/java/technologies/javase/desktop-overview.html) and servers. Java offers the rich user interface, performance, versatility, portability, and security that today's applications require..

**Core Java – OOP using Java, Programming Fundamentals using Java, Advanced Java – Multithreading, Error or Exceptional Handling, Collections Frameworks, JDBC, Swings and Applet.**

**JEE –** **Java EE** is a structured application with a separate client, business, and **Enterprise** layers. It is mostly **used** to develop APIs for Desktop Applications like antivirus software, game, etc. It is mainly **used** for developing web applications.

**Eg: Servlets, JSP, EJB, RMI,JSF**

**API :** Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

**JME :** Java Platform, Micro Edition (Java ME) provides a robust, flexible environment for applications running on embedded and mobile devices in the Internet of Things(IOT).

1. Difference Between an Object and Class?

Ans :

|  |  |
| --- | --- |
| Object | Class |
| Object is an **instance** of a class. | Class is a **blueprint or template** from which objects are created. State & Behaviour |
| Object is a **real world entity** such as pen, laptop, mobile, bed, keyboard, mouse, chair etc. | Class is a **group of similar objects**. |
| Object is created through **new keyword** mainly e.g. Student s1=new Student(); | Class is declared using **class keyword** e.g. class Student{} |
| Object is created **many times** as per requirement. | Class is declared **once**. |

Some Additional points on Class :

* A Class either be public or default
* .java file can have only and only one public class.
* There can be any number of default classes in .java file but there will be only one public class in java.

1. What is the root class from which every class extends?

Ans: Object Class

1. What are the primitive data types in Java?

Ans: 8 data types and its size

(whole numbers)

1. byte 1-byte default value 0;

2. short 2-bytes default value 0

3. int 4-bytes default value 0

4. long 8-bytes default value 0L. must put L at the end otherwise considered as int.

(decimal values are larger than whole numbers)

5. float 4-bytes default value 0.00000f

6. double 8-bytes default value 0.00000f(any decimal values in java by default considered as double)

7. char 2-bytes '\u0000' why 2 bytes? : It supports 1-byte(ASCIII) and 1-byte(UNICODE) = 2-bytes.

8. boolean 1bit false by default/true

1. Where are Strings stored?

Ans:

**Maven** is a powerful project management tool that is based on POM (project object model). It is **used** for projects build, dependency and documentation. It simplifies the build process.

A Project Object Model or **POM** is the fundamental unit of work in Maven. It is an **XML** file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

It Connects it to remote repo where all the dependencies will be available, from that repo maven download and installed it on to the client machine. It won’t installed it blindly rather it is tested and veried then only it gets installed on the client machine.

Dev environments : Devleopers machine, Writes code.

Production Environment : project live now where customer can access it.

.m2 file : Whatever we ask maven to download it, it downloads from the repo and stored it into the local folder that it is .m2 folder in C drive. Why it does that? If I need same repo it does not have to download it from the remote repo again and again.

Steps creating Maven Build : New => other => maven => Maven Project

Step 2: tick that archive it gives predefined structure we no need it actually.

Step 3: GroupId : Naming our Root Package name of your app. ArtifactId : same as project Id.

GroupId : com.app eg : com.app.applicationlayer

ArtifactId : FirstMavenProject same in name also.

src/main/java = where all your java classes will go .java

src/main/resources = any config file or .properties file (.xml files) for ur java app

src/main/test = All java test cases files goes here.

src/main/resources = All config files like .properties files

step 5: select POM.xml add all the dependencies then right clik on pom.xml => maven => update project

Step 6 : change jre to jdk

Step 7 run as maven build

Step 8 refresh and look for archive folder you wll find snapshot.jar right clik on that then open with git bash

Step 9 : java -cp projectname com.app.Main then enter

Step 10 : ouput HELLO MAVEN

1. What are collections in Java?

Ans: The Java Collection provides an architecture to deal with the group of objects. The collection means a single unit of objects. It allows us to store and manipulate the group of objects as a single unit.

We can easily perform many operations such as searching, sorting, insertion, deletion, and manipulation on a group of objects using the collection framework.

Java Collection framework provides several interfaces and classes.

The interfaces contain Set, List, Queue, Deque, and classes contain ArrayList, Vector, LinkedList, HashSet, TreeSet, LinkedHashSet, and PriorityQueue.

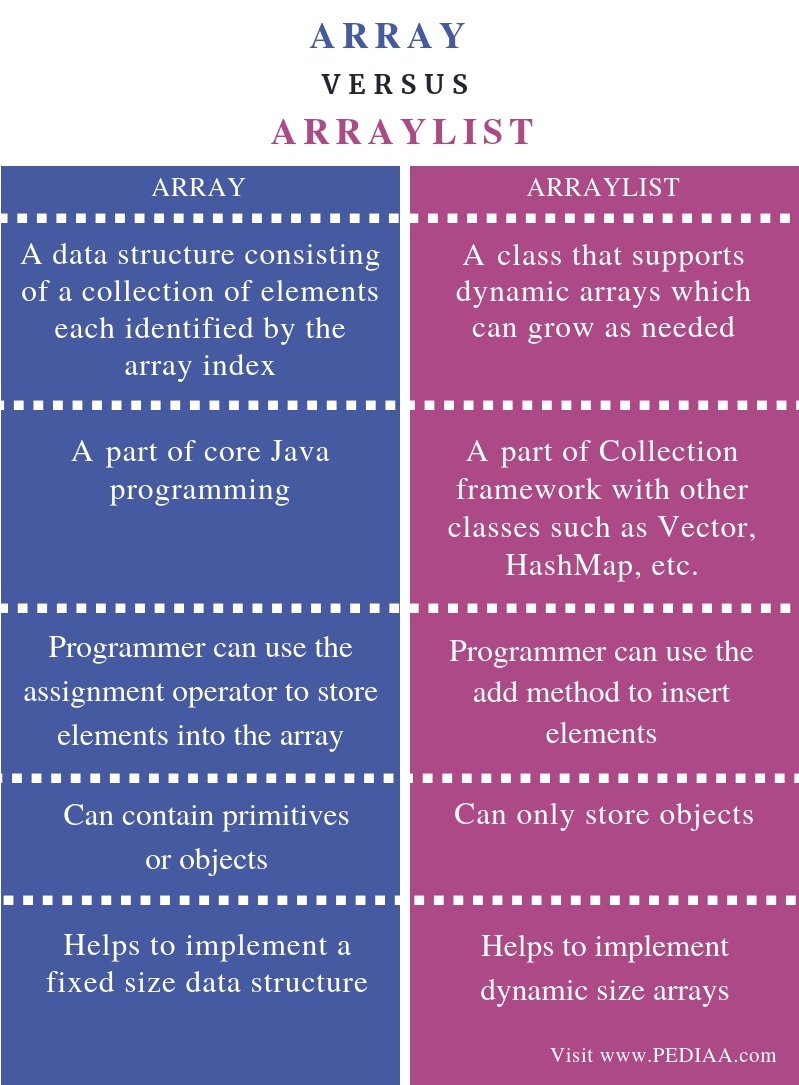
1. What are the interfaces in the Collections API?

Ans : Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) .

1. What is the difference between a Set and a List?

Ans:

1. What is the difference between an Array and an ArrayList?

Ans: