**Java Self-notes**

**1] Java ( Basics of Programming ):**

1. **Introduction of Java:-** Java was developed by James Gosling, Patrick Naughton and Mike Sheridan at Sun microsystem in 1991.In 1991 initial name of Java was Oak but it was TM of another brand so it renamed as a Java in 1995.
2. **Major application of Java:-** Desktop applications, android applications ,web applications like (LinkedIn, Snap deal), Embedded system. Robotics and games.
3. **Check java is installed in your in computer or not using command:** Java -- version on command prompt.
4. **Features of Java:-**
5. **Java** is easy to learn. Concepts of C++ like pointers and operator overloading are removed in Java.
6. Java is OOP.
7. In Java, both compile time and run time error checking performed. Java mainly focuses on memory management and exception handling by introducing the feature of automatic garbage collector.
8. **Java** is **platform independent**. Here meaning is that in Java, we can make code in one machine and can run it on any other machine by using JDK. Why? In java when we compile code it is converted into **Byte code** and this code is platform independent and it is compatible to any other OS or machine but only condition is that on that machine JDK must be installed.
9. **Java** is most secure programming language because java program runs in JRE with almost null interaction with system OS.
10. **Multithreading in Java:** Java program can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory to execute multiple threads at the same time. **E.g.: While typing some text, grammatical errors are checked along with our typing like in MS word.**
11. **Java program is architectural neutral. Java (.java) => byte code(.class).**
12. **Difference between C++ and Java:**

|  |  |
| --- | --- |
| **C++** | **Java** |
| 1. Platform dependent. | 1. Platform independent. |
| 1. Supports Operator overloading, pointers. | 1. Does not support pointers and op. overloading. |
| 1. It uses a compiler only. | 1. It uses both compiler and interpreter.(bytecode/.class) |
| 1. Runtime error detection is not present in the system. | 1. Runtime error detection is handled by system. |
| 1. New and delete keyword are used to manage objects. | 1. **Automatic Garbage collector** is used for object management. |

1. Difference between JVM, JRE and JDK:

|  |  |  |
| --- | --- | --- |
| **JVM** | **JRE** | **JDK** |
| 1. Java virtual machine is a machine that provides runtime environment to execute byte code. 2. Java virtual machine does not understand .java code. 3. Java virtual machine understands byte code. | 1. Java runtime environment is combination of JVM and different libraries and other components that are required in applications. 2. Java runtime environment does not contains tools like compiler and debugger. | 1. Java development kit is superset of JRE and it contains all tools like compilers , debuggers and components. |

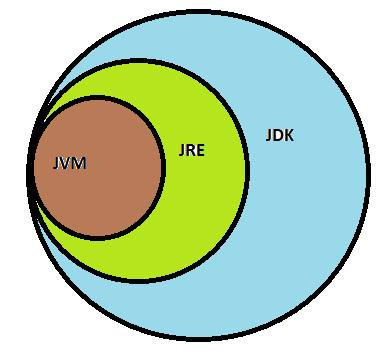


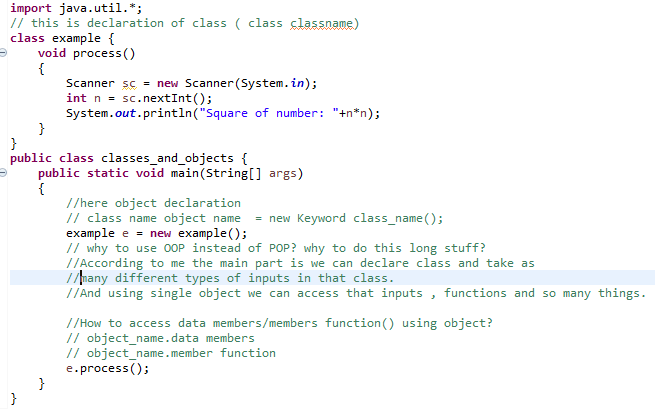
Figure 1 JDK > JRE > JVM

1. **KILPO/KILSO (Keywords, Identifiers, Literals/constants, Punctuators/Separators, Operators)**
2. **4 main parts of Java code:**
3. Class
4. Public static void main(String[] args)
5. Static: Use of static is to make void main function static. We know that Java program starts with class and ends with class. So inside function to access function we need to call function by using object. But after using static keyword we need not to create object and we can call that function only once and this call made by JVM.
6. main() method: Program execution starts with main().

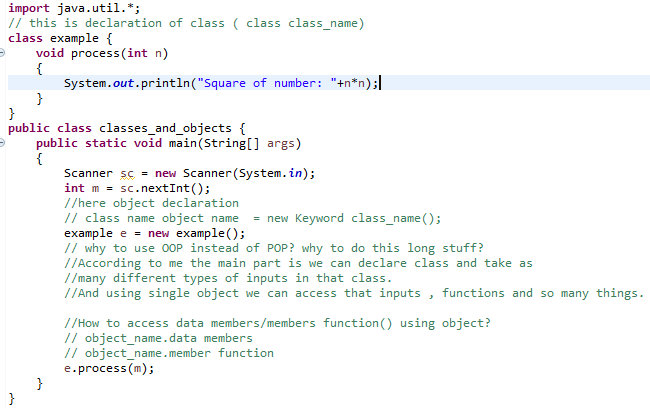
**2] Java OOP concept:**

**1. Classes and Objects:**

**i. Understand example of classes and object:**

****

**ii. Passing argument using object**

****

**iii. What is difference between normal variable and static variable (specifically in OOP)?**

**Ans:** 1) Static data members are not part of any object. They are part of classes. But all objects can access that value.

**2)** When we declare normal variable in a class and once it’s object is created then according to data type of that variables memory is allocated for that object.

3) As static variables are not part of object so it is unique and its memory is not included in the object area so we can save the memory.

**4) Use: Why to use static?**

Suppose we have to enter students data and we declare roll no, names and college names.

If we declare college name in normal variable form then for every object college name data memory is repeated which is wastage of memory and here is static comes in use. So use static for college name(Visualise it).

**iv. What are static member function?**

**Ans:** 1) Static members are not part of objects they are part of classes.

2) They can be invoked directly using class name.

3) Static member functions contain only static variables and normal var are not allowed in that.