# JAVA PROGRAMMING(MCA202)

1. **BASICS OF JAVAs**

**What is JAVA?**

* Java is a platform independent programming language.
* Java is a High-level,class based Object oriented programming(OOP) language.
* It is designed to be platform-independent and follows the “Write Once,Run Anywhere”(WORA) philosophy.

**Where is JAVA used?**

* Java is used in various domains,including web development,mobile application development(Android),enterprise applications,scientific applications,and more.

**Why use JAVA ?**

1. Platform Independence: Java programs can run on any device that supports Java.
2. Object-Oriented: Encourages modular and reusable code through the use of classes and objects.
3. Robust: Java Includes features like strong memory management,exception handling,and garbage collection.
4. Multi-threaded: Java Supports concurrent execution of multi-threads , improving program efficiency.

**History of JAVA ?**

* Java was developed by James Gosling and his team at Sun Microsystems in 1995.
* Java is now owned by Oracle.
* Firstly, it was called ****"Greentalk"****.After “**OAK**”.
* Then in 1995 “**OAK**” was renamed as “**JAVA**”.

**Features of JAVA ?**

1. Simple: Easy to learn and use.
2. Platform Independent: Runs on any device with a Java Virtual Machine(JVM).
3. Object-Oriented: Everything in Java is an Object.
4. Robust: Features like strong memory management,exception handling,and garbage collection.
5. Multi-threaded: Supports concurrent execution of multi-threads.
6. Distributed: Supports networked and distributed computing.
7. Secure:Provides a secure execution environment.
8. Architecture-Neutral: No implementation-dependent aspects.

**Internals of JAVA Program ?**

* A Java program is written in plain text and save with a .java extension.
* The Java compiler(javac) translates the source code into byecode(.class files).
* The Java Virtual Machine(JVM) executes these byte codes on the target platform.

|  |
| --- |
| * Java is a case-sensitive language. * Every line of code that runs in java must be inside a class. * A class should always start with an uppercase first letter. * The name of the Java file should match the class name. |

**Compiling and Executing Java file:**

|  |
| --- |
| import java.io.\*;  class First  {  public static void main(String args[])  {  System.out.println("Welcome to JAVA PROGRAMMING...");  }  } |

* **First.java**

1. **Open CMD and change directory:**

|  |
| --- |
| Microsoft Windows [Version 10.0.19045.3803]  (c) Microsoft Corporation. All rights reserved.  C:\Users\marsh>e:  E:\>cd MCA\SEM 2\JAVA PROGRAMMING (MCA202) |

1. **Compile Java file:**

|  |
| --- |
| Microsoft Windows [Version 10.0.19045.3803]  (c) Microsoft Corporation. All rights reserved.  C:\Users\marsh>e:  E:\>cd MCA\SEM 2\JAVA PROGRAMMING (MCA202)  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>javac First.java  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

1. **Execute Java file:**

|  |
| --- |
| Microsoft Windows [Version 10.0.19045.3803]  (c) Microsoft Corporation. All rights reserved.  C:\Users\marsh>e:  E:\>cd MCA\SEM 2\JAVA PROGRAMMING (MCA202)  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>javac First.java  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>java First  Welcome to JAVA PROGRAMMING...  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

**So,execution of Java program is a Two stage process :**

1. **Compilation:**Java compiler(javac) translates the source code into byecode(.class files).
2. **Interpretation:** Converts these byte code into binary.

**Compiler VS Interpreter :**

|  |  |
| --- | --- |
| **Compiler** | **Interpreter** |
| 1. Compiler will analyze the program and check their correctness.  * If an error is found ,it throws an error message. * If the program contains no error,then the compiler will convert the source code into machine code. | 1. The source code statements are executed line-by-line during their execution.  * If an error is found at any specific statement ,then the interpreter stops further execution until the error gets removed. |
| 1. It links all the code files into a single runnable program ,which is known as .exe file.And runs the program and generates the output. | 1. No linking of files happens,or No machine code will generate separately. |
| 1. A compiler translates complete High-level programming code into machine code at once. | 1. An interpreter translates one statement of programming code at a time into machine code. |

**Difference between JDK,JRE and JVM :**

|  |  |  |
| --- | --- | --- |
| **JDK** | **JRE** | **JVM** |
| 1. Java Development Kit(JDK) is a software development kit to develop applications in Java.In addition to JRE,JDK also contains number of development tools(compilers,JavaDoc,Java Debugger etc.). | 1. Java Runtime Environment(JRE) is the implementation of JVM and is defined as a software package that provides Java class libraries along with JVM. | 1. Java Virtual Machine(JVM) is an abstract machine that is platform-dependent and has three notions as a specification:    1. A document that describes requirement of JVM implementation    2. A computer program that meets JVM requirements.    3. An implementation that executes Java byte code provides a runtime environment for executing Java byte code. |
| 1. JDK = JRE + Development tools. | 1. JRE = JVM +Libraries to run the application. | 1. JVM = Only Runtime environment for executing the Java Byte code. |
| 1. JDK includes tools for developing,compiling,and debugging Java applications. | 1. JRE provides necessary runtime libraries for executing Java applications. | 1. JVM executes Java byte code on target platform . |

**Internal details of JVM :**

* Jvm is responsible for loading,interpreting,and executing Java bytecode.
* It includes components like Class Loader,Bytecode Verifier,Interpreter,Just-In\_time(JIT) and Garbage Collector.
* JVM provides following operation:

1. Loads code.
2. Verifies code.
3. Executes code.
4. Provides runtime environment.

|  |
| --- |
| jvm-3 |

**Variable and Datatype :**

* A Variable is a container which holds the value while Java program is executed.
* A variable is assigned with a data type.
* Variable is a name of memory location.
* There are 3 types of variables in Java:

1. Local : A variable declared inside the body of the method is called local variable.A local variable cannot be defined with “static” keyword.
2. Instance : A variable declared inside the class but outside the body of the method is called instance variable.It is not declared as static.
3. Static : A variable declared as static is called a static variable.It cannot be local.You can create a single copy of the static variable and share it among all instance of the class.Memory allocation of static variables happens only once when the class is loaded I the memory.

* There are 2 types of Data types in Java:

1. Primitive : Boolean,char,byte,short,int,long,float and double.
2. .Non-primitive: Classes,interfaces and Arrays.

|  |
| --- |
| 1. **public** **class** A 2. { 3. **static** **int** m=100;//static variable 4. **void** method() 5. { 6. **int** n=90;//local variable 7. } 8. **public** **static** **void** main(String args[]) 9. { 10. **int** data=50;//instance variable 11. } 12. }//end of class |

|  |
| --- |
| java-data-types |