

Experiment No.1

Aim: Study and implement basic concepts of Java Programming Language.

Problem Statement:

1. Write a Java program to display student result and its grade using if-else-if statement.
2. Write a Java program to perform arithmetic operations on 3x3 matrices using multidimensional array.
3. Write a program to perform basic arithmetic operations using switch statement.

Theory:

History of Java

- Java was developed by Sun Microsystems in the year 1991.
- Later it is acquired by Oracle Corporation. It is now subsidiary of Oracle.
- Java is developed by James A. Gosling and Patrick Naughton.
- Before Java, its name was Oak.
- Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

What is Java?

Java is a general-purpose computer-programming language. Java is a programming language and a platform. It is object-oriented programming language similar to C++. It is an object-oriented language similar to C++, but with advanced and simplified features. Java is free to access and can run on all platforms.

Java is: –

- Concurrent where you can execute many statements instead of sequentially executing it.
- Class-based and an object-oriented programming language.
- Independent programming language that follows the logic of “Write once, Run anywhere” i.e. the compiled code can run on all platforms which support java.

In simple words, it is a computing platform where you can develop applications.

Platform:

Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

Features of Java language –

- **Simple** – Java was designed to be easy for the professional programmer to learn and use effectively. Java is very easy to learn, and its syntax is simple, clean and easy to understand.
- **Object-oriented** –Everything in Java is an object. Object-oriented means we organize software as a combination of different types of objects that incorporate both data and behavior.

- **Platform Independent**
Java is platform independent. A platform is the hardware or software environment in which a program runs. Java code can be executed on multiple platforms. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere (WORA).
- **Robust** – The multi-platformed environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. Thus, the ability to create robust programs was given high priority in the design of Java.
- **Multithreading** – Java supports multithreaded programming which allows you to write programs that do many things simultaneously.
- **Architecture-Neutral** – Java program execution is not depends on OS upgrades, processor upgrades and changes in core system resources.
- **Interpreted and high-performance** – Java enables the creation of cross- platform programs by compiling into an intermediate representation called Java bytecode. This code can be interpreted on any system that provides a Java Virtual Machine. Java bytecode was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using Just-in-time compiler.
- **Distributed** – Java is designed for distributed environment of the Internet, because it handles TCP/IP protocols.
- **Dynamic** – Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run-time.

Java platform components

- Java platform components are:
 - Java language
 - The Java Development Kit (JDK)
 - The Java Runtime Environment (JRE)
 - The Java Compiler(Javac)
 - The Java Virtual Machine (JVM)
- Java platform also contains
 - Garbage collectors,
 - Set of libraries and
 - Additional components and tools that are required to efficiently run the Java applications.

Simple Java Program:

```
class Simple
{
    public static void main(String args[])
    {
        System.out.println("Welcome to First program");
    }
}
```

- Name of the file must be same as the name of the class.
- The file extension is *.java*.
- Java is *case-sensitive*.
- To compile java program use—
javac filename.java
- To execute java program use —
java filename
- `main()` must be declared as **public** since it must be called by the code outside of its class when the program is started.
- **static** keyword allows `main()` to be called without having to instantiate a particular instance of the class.

Data-types—

- Defines types of data used in program.
- Java defines eight primitive types of data:
 - byte, short, int, long, char, float, double, and boolean.

<i>Data-type</i>	<i>Size</i>
Byte	1-byte
Short	2-byte
Int	4-byte
Long	8-byte
Float	4-byte
double	8-byte
character	2-byte
boolean	1-bit

Array—

- Array is a group of same type variables referred by common name. An array is a group of like-typed variables.
- An array is a collection of similar type of elements.
- The elements of an array are stored in a contiguous memory location.

- It is a data structure where we store similar elements.
- Array in Java is index-based. A specific element in an array is accessed by its index.
- The variables in array are ordered & each has an index beginning from 0.

Types of array:

- Single/One Dimensional Array
- Multidimensional Array

1. One dimensional array–

- A one-dimensional array is a list of like-typed variables.
- To create an array, first must create an array variable of the desired type.
- **Array declaration:**

`type var-name[]; OR type[] var_name;`

- **Example-** `int month_days[];`
- **Memory allocation for array–**

`var-name=new type[size];`

2. Multi-dimensional array–

- It is used when data is stored in row and column based index
- It is also known as matrix form.
- In Java, multidimensional arrays are actually arrays of arrays.
- To declare a multidimensional array variable, specify each additional index using another set of square brackets.
- **Declaration:**

`type var-name[][]=new type[rows][columns];`

- **Example**

`int number[][] = new int[4][5];`

Conclusion:

Students are able to understand basic concepts of java programming language and write simple java programs.