Think and Tell

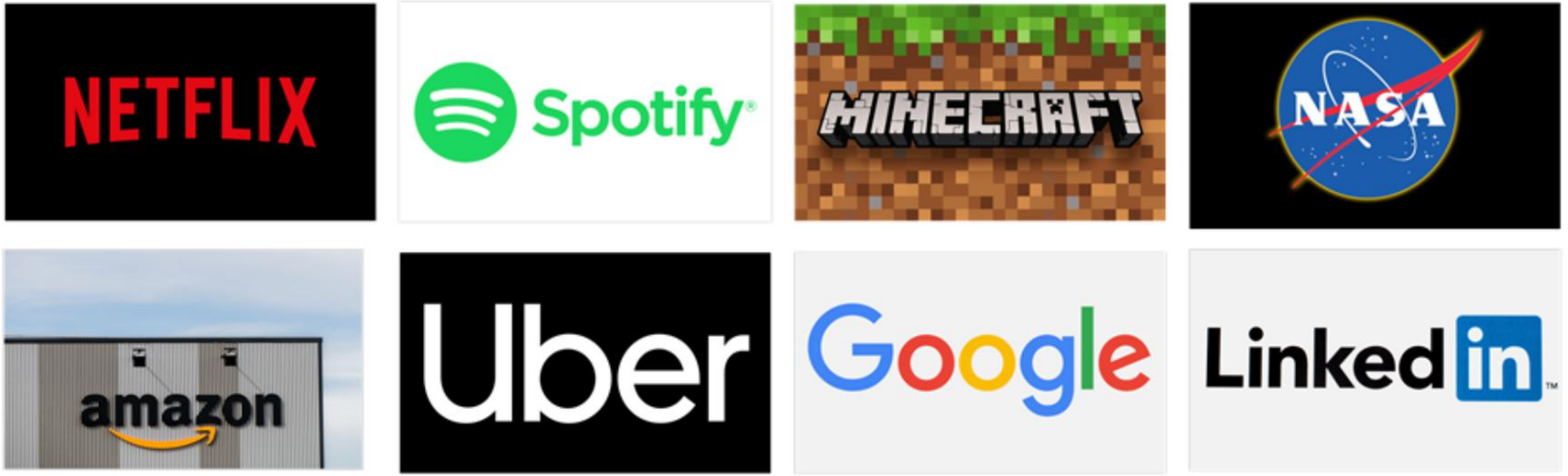
- Are you an Android user?
- As a software developer, what are your chances of encountering a Java program?
- Can a software developer work without knowing an object-oriented programming language?

Java Is Everywhere

Amazon, the world's largest e-commerce website, provides a seamless online shopping experience to its customers.

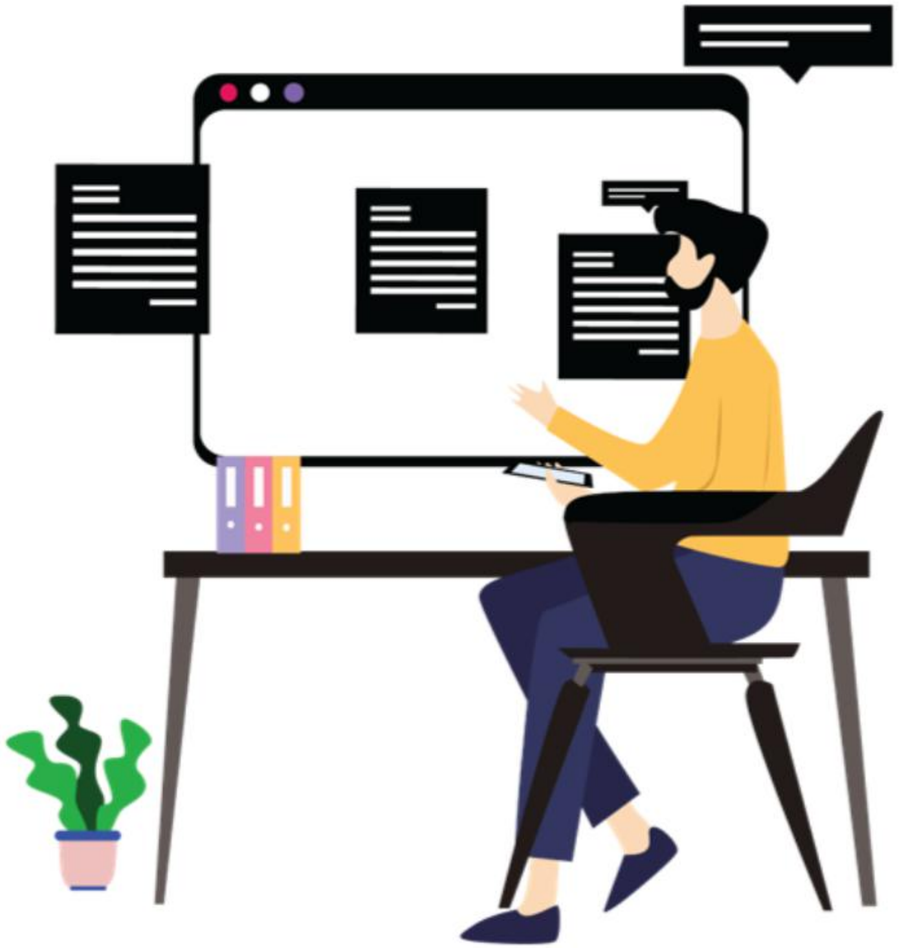
Do you know on which platform it is created?

What Are the Benefits of Learning Java?



Which language is used by these companies to manage their platforms and applications?

Introduction to Java, Variables, Datatypes, and Operators



Learning Objectives

- Introduction to Java
- Deconstructing a Java Program
- Explain the Syntax and Semantics
- Define Datatypes
- Declare and Initialize Variables
- Use Arithmetic and Relational Operators
- Declare Expressions
- Define Statements

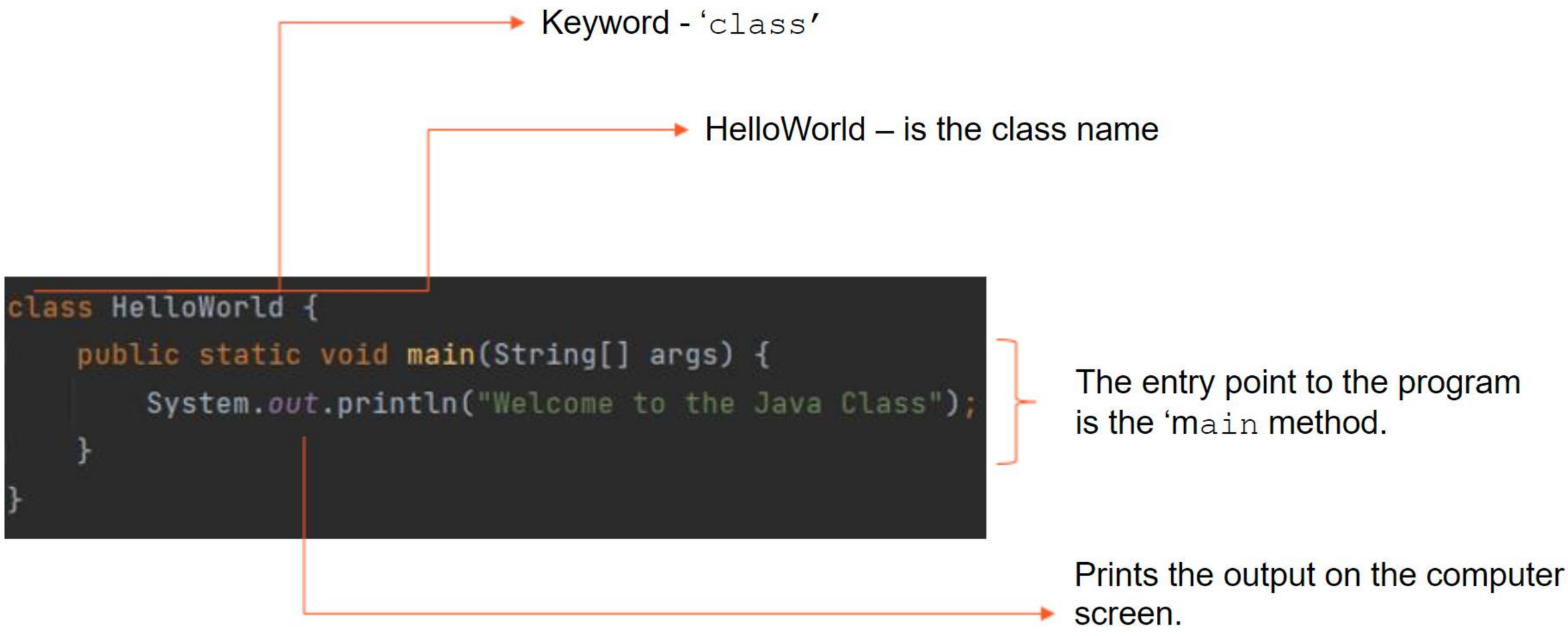
Introduction to Java

- Java is a high-level programming language that is used to build applications.
- We can write programs using Java that are simple and readable.
- Programs are a set of instructions given to the computer to perform various tasks.
- Java programs can be used to build web applications, mobile games, etc.

Deconstructing a Java Program

How Does a Simple Java Program Look Like?

- A simple Java program that prints `'Welcome to the Java Class'` is shown below,



Note : Keywords and the main method will be discussed in detail in further sessions

Components of a Java Program

- Class
  - All Java programs consist of at least one class.
  - The instructions or lines of code are written inside a class.
  - A class is defined by a name, in this case, `'HelloWorld'` is the class name.
- Main method
  - A method is a member or part of a class.
  - The main method is the entry point of any Java program.

Where Can You Write Java Programs?

- A Java program can be written on any:
  - Text Editor or Notepad
  - IDE
  - Any web-based IDE (Integrated Development Environment)
- The file must be saved with the **.java** extension.
- The name of the file and the class name must be the same.
- Java program with the **HelloWorld class** is saved as **HelloWorld.java.**

How Does a Computer (OS) Understand a Program?

- The OS must be able to identify the saved **.java** file.
- To do that, install Java on your machine.
- Once the installation is done, the OS recognizes the files saved with the **.java** extension.
- Note that a Web IDE Java environment is integrated into the IDE itself.

How Can You Execute a Java Program to Get the Desired Output?

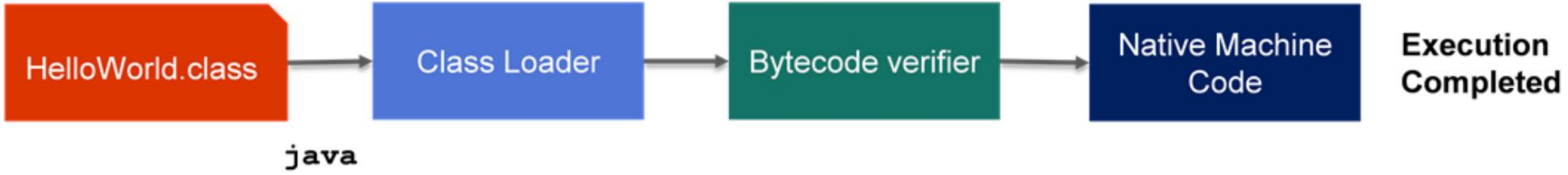
- A set of instructions in the Java program must be run or executed on the computer to get the expected output.
- The Java program goes through a 2-step execution process:
  - Compilation
  - Execution

Compilation of the Java Program

- Java program in the text format is converted to an executable format called the **byte code**.
- The `javac` command, compiles the Java program.
- Upon compilation, a bytecode is generated that is saved as a **.class** file in the system where the **.java file is present**
- **HelloWorld.java** generates a **HelloWorld.class** file which is the bytecode.

Execution of the Java Program

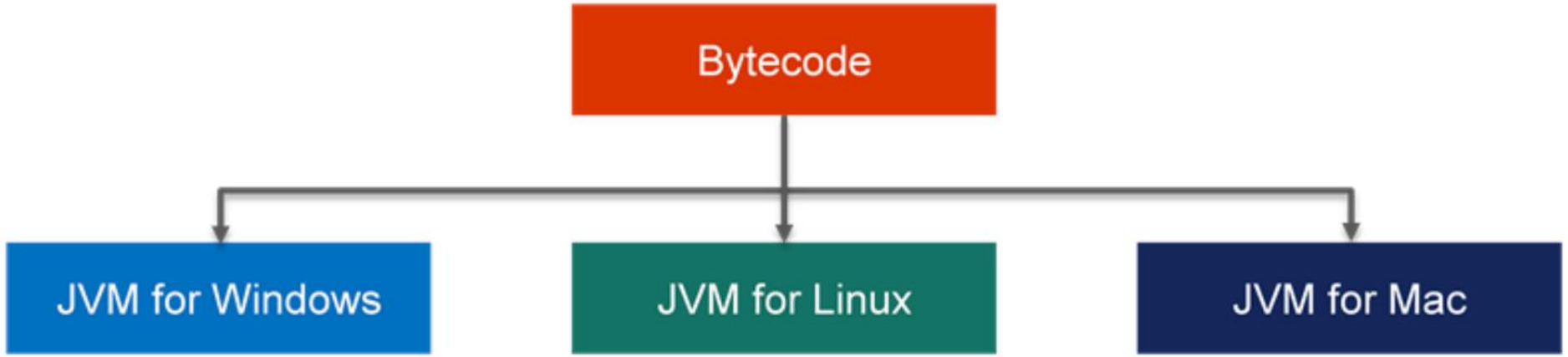
- The **bytecode** or the **.class** file must be executed to give the desired output.
- The **.class** file is passed to a class loader that interprets the class file as a machine code readable by the computer.
- A byte code verifier validates the **.class** file.
- Finally, the bytecode is converted to a machine-readable code.
- The `java` command is used for this.



Note: *The filename and the class name must be the same*

Key Features of Java – WORA

- The byte code generated during the compilation process is platform independent and can be executed on any OS (Windows, Linux, or Mac).
- The JVM is platform dependent i.e., each OS will have its implementation of the JVM.
- The JVM converts the byte code to machine code understandable by the underlying OS.
- Thus, a program written in Java need not be modified when executed on multiple OS, hence Java is called as Write Once Read Anywhere(WORA).



Quick Check

The _____ is the entry point to a Java program.

1. class
2. keyword
3. main method
4. bytecode

Quick Check: Solution

The _____ is the entry point to a Java program.

1. class
2. keyword
3. **main method**
4. bytecode

Java Syntax

- Syntax in computer programming refers to the rules that control the structure of symbols, punctuation, and words of a programming language.
- Syntax defines the meaning or the semantics of a programming language.
- If a Java program does not follow a certain syntax, the compiler will not understand the program.
- A compiler reading an incorrect syntax will not allow the code to be compiled.

Basics of Java Syntax

- All lines of code must be present inside a class and end with a semicolon “;”
- The main method must be written in a format as shown below,

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Welcome to the Java Class");
    }
}
```

- Standard naming conventions must be followed for identifiers
  - Identifiers are the names given to the elements of a Java program.
  - A class is an element of a Java Program and has a name associated with it.
  - The class name is the **identifier** i.e., `HelloWorld`.
- A set of reserved words called as keywords must be used.
  - For example: `class`, `public`, `static` are keywords.

Note : Keywords will be discussed in detail in further sessions

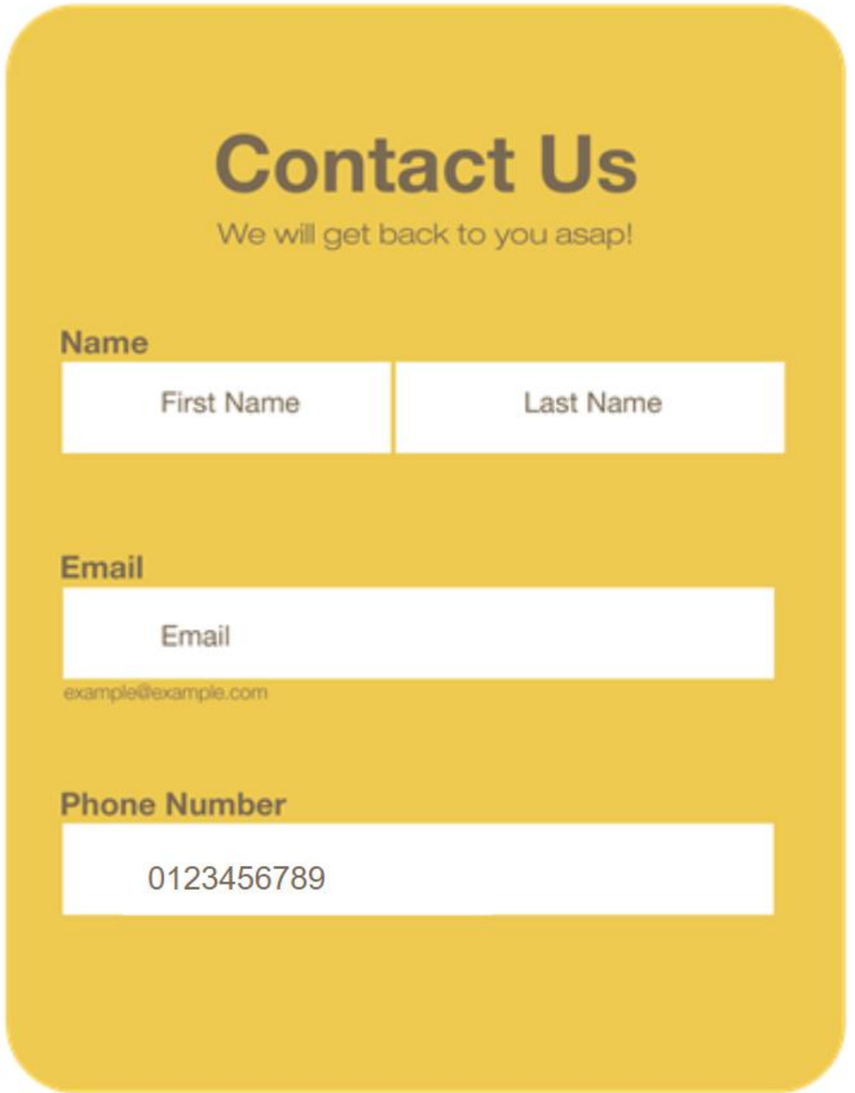
Hello World

Write a Java program to display ‘Hello World’ on execution.
Use the Web IDE to do the same.

DEMO

Which Type of Data Will Be Stored in This Form?

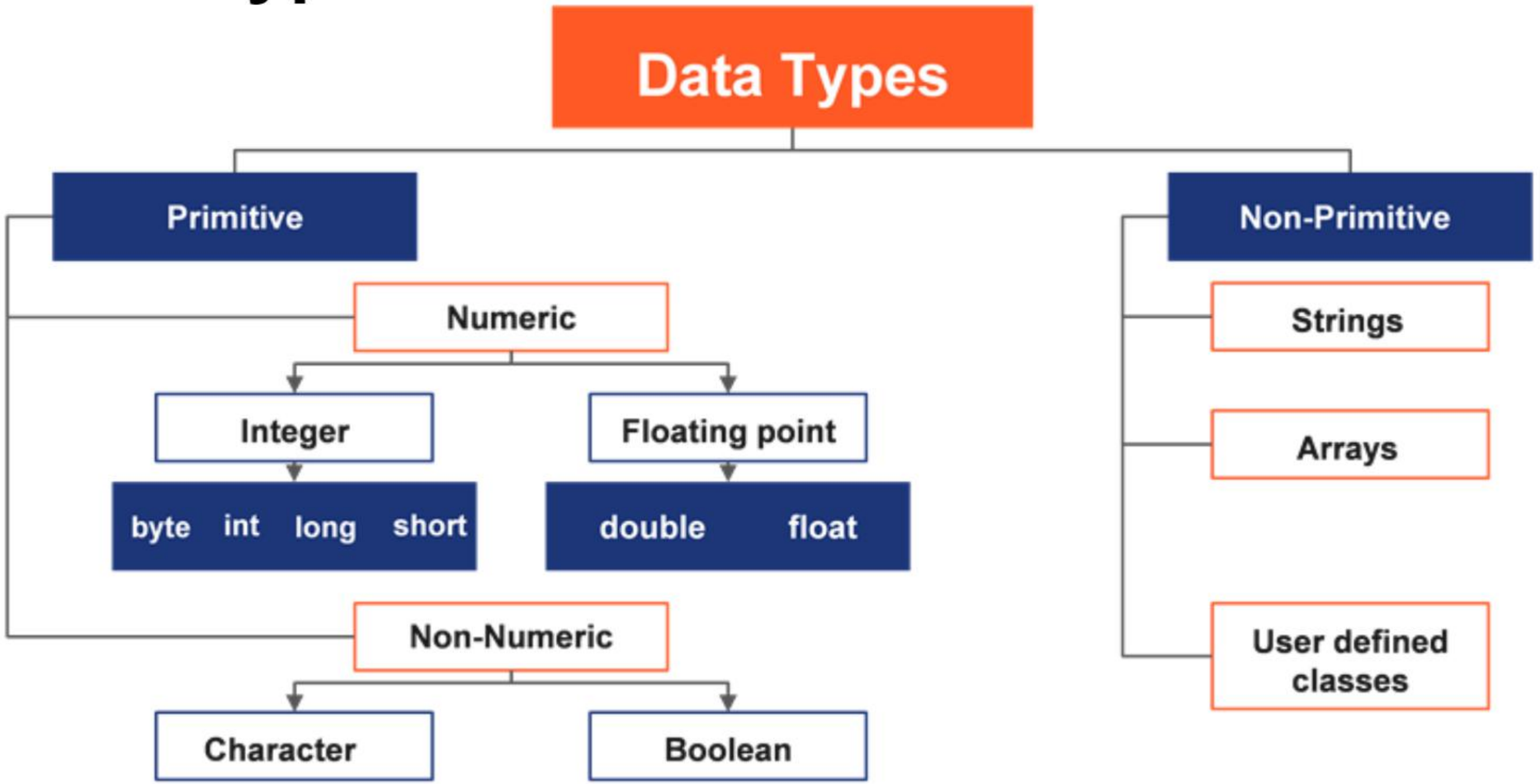
There are numeric and non-numeric values in the form. How can these values be saved in any application?



Data Types

- Data types refer to the type of data that we can store inside a program.
- Java provides support to store multiple data types like integer, decimal, boolean, character and string.
- Data types in Java are classified into two categories:
  - Primitive Data Types – These are built into the core of the Java language.
  - Non-primitive Data Types – These are external types that provide support to store a certain kind of data.

Java Data Types



- Built-in data types in Java are known as the primitive or the simple data types.
- Non-primitive data types are not pre-defined in the Java language.
- When a non-primitive data type is used it references a memory location where the data must be stored. They are also called Reference Datatypes.

Quick Check

Identify the data types used in this form.

Contact Us

We will get back to you asap!

Name

Steve | Walker

Email

steve.walker89@gmail.com

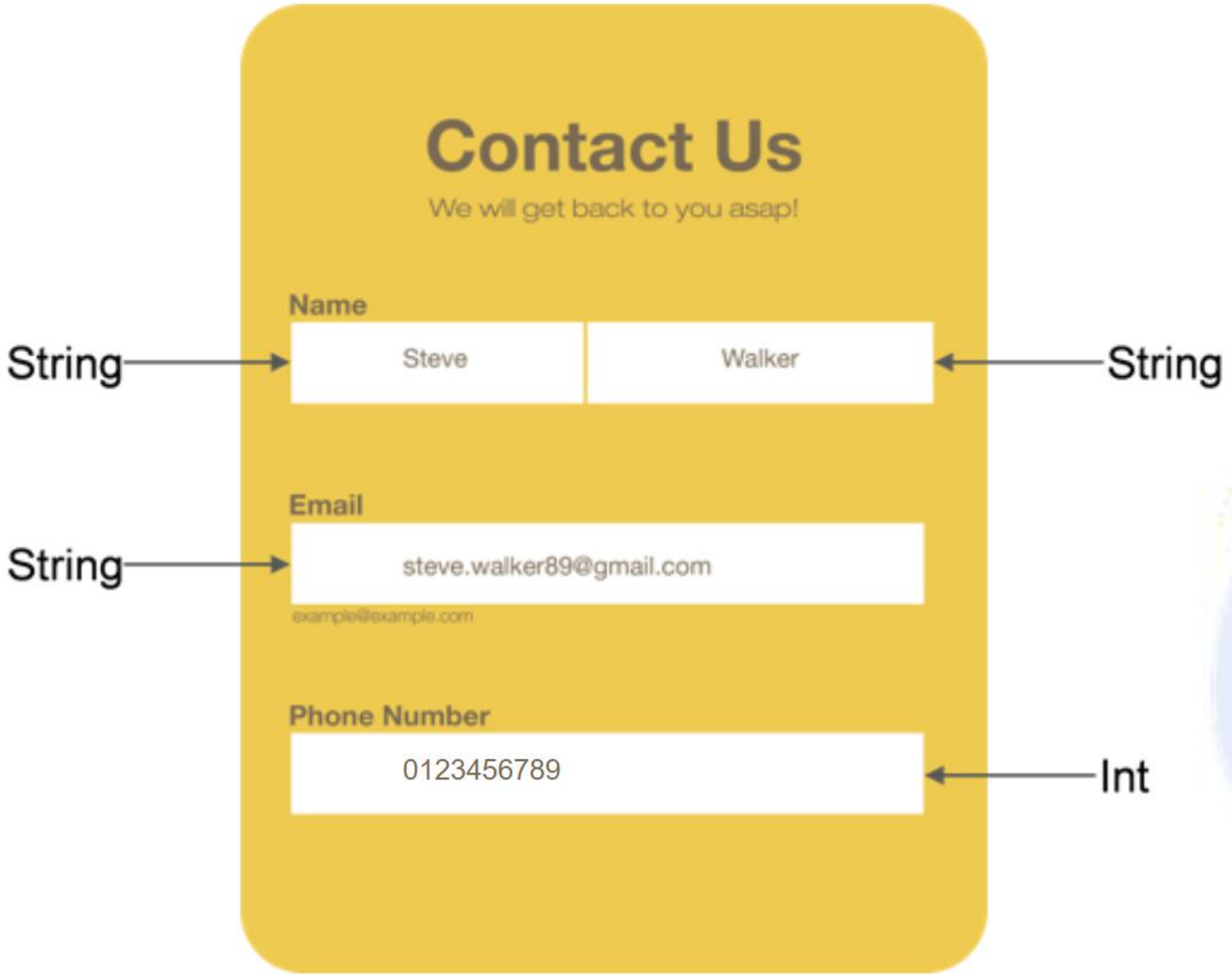
example@example.com

Phone Number

0123456789

Quick Check – Solution

Identify the Data Types used in this form.



Variables

- A variable is a container used to hold data belonging to a specific data type.
- Java allocates memory to each variable used in a program according to the datatype.

Variable	Memory

- Each variable used in a program must be declared.
- The syntax to declare a variable with a specific datatype is

```
<data-type> <variable name>;
```

Datatypes	Memory allocated in bits
byte	8
short	16
int	32
long	64
float	32
double	64
char	16
boolean	1

Declaring Variables

- Declaring an integer variable called **number**

```
int number;
```

- Declaring a decimal or floating-point variable called **decimalNumber**

```
float decimalNumber;
```

- Declaring a char variable called **character**

```
char character;
```

- Declaring a Boolean value called **bool**

```
boolean bool;
```

- Declaring a String value

```
String str;
```

Naming Conventions for Variables

- The name of a variable needs to be meaningful, short, and without any embedded space or symbol.
- A variable name must begin with a letter, an underscore (_), or the dollar symbol ($), which can be followed by a sequence of letters or digits (0 to 9), '$', or '_'.
- A variable name should not:
  - Start with a digit.
  - Contain embedded white spaces.
  - Be a keyword.
- A variable name in Java is case-sensitive.

Valid Variable Names

```
int _rollNo;
int $;
int $1234;
int _4567;
float salary;
```

Invalid Variable Names

```
int 1rollNo;
int roll No;
float class;
```

Declaration of Variables

- In the class called Calculation, the variables are declared inside the main method.

```
class Calculation {
    public static void main(String[] args) {
        int number;
        float decimalNumber;
        char character;
        boolean bool;
    }
}
```

Initializing Values to Variables

- Initialize an integer variable called **age** to the value 20

```
int age = 20;
```

- Initialize a float variable called **salary** to the value 20.45

```
float salary = 20.45f;
```

- Initialize a char variable called **gender** to the value ‘m’

```
char gender = ‘m’;
```

- Initialize a boolean variable called **isEmployed** to the value true

```
boolean isEmployed = true;
```

- Initialize a String variable called **name** to the value John

```
String name = “John”;
```

The values assigned to the variables are called **literals**.

A few types of literals are

1. Integer literals
2. Float literals
3. Double literals
4. Character literals
5. Boolean literals
6. String literals

The **assignment operator ‘=‘** is used to assign values to the variable.

Float and Long Literals

- Java provides `int` and `long` data types to store integers.
- The `float` and `double` data types are used to store decimal datatype.
- Java enforces that float literals be suffixed with `'f'` and long literals be suffixed with `'L'` during initialization to avoid confusion.
- `float` and `double` initialization

```
float floatNumber = 25.89f;
double doubleNumber = 34.7;
```

- `int` and `long` initialization

```
int number = 10;
long longNumber = 35L;
```

Initialization of Values to Variables

- There are two ways to initialize values to variables:
  - **Declare first, Initialize later** – Variables can be declared and then values can be assigned to them.
  - **Declare and Initialize** – Variables can be initialized during declaration itself.

Declare first, Initialize later

```
class Calculation {
    public static void main(String[] args) {
        int number;
        number = 10;
        long longNumber;
        longNumber = 35L;
        float floatNumber;
        floatNumber = 25.89f;
    }
}
```

Declare and Initialize

```
class Calculation {
    public static void main(String[] args) {
        int number = 10;
        long longNumber = 35L;
        float floatNumber = 25.89f;
        double doubleNumber = 34.7;
        char character = 'j';
        boolean bool = false;
    }
}
```

Displaying Output on the Console

- The `System.out.println` method displays the output on the console.
- Any value given within the double quotes “ ” will be displayed on the console.

```
System.out.println("Welcome to the Java Class");
```

- The console is a window of the operating system through which users can interact with system programs.
- The interaction consists of text input from the standard input (keyboard) or the text displayed on the standard output (computer screen).

Employee Details

Write a Java Program to store details like the name, age and monthly salary of an employee.

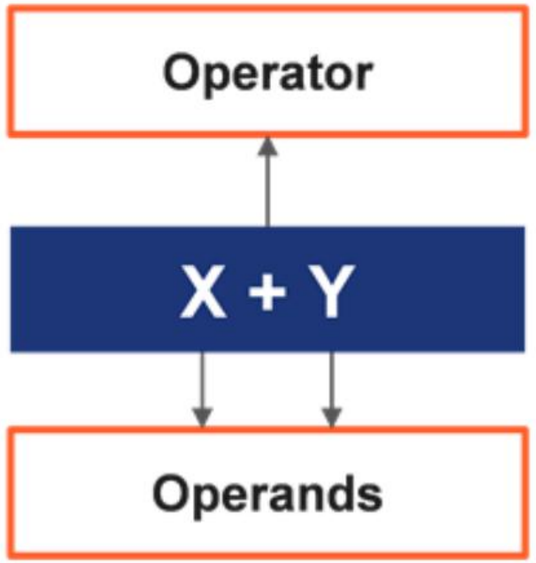
- Use the Web IDE to write the program.
- Write the program inside a class called **EmployeeDetails.**
- Declare variables and initialize the values for name, age and salary inside the main method using the table given below.

Variable Name	Value
name	Larry
age	28
monthlySalary	$2500.78

DEMO

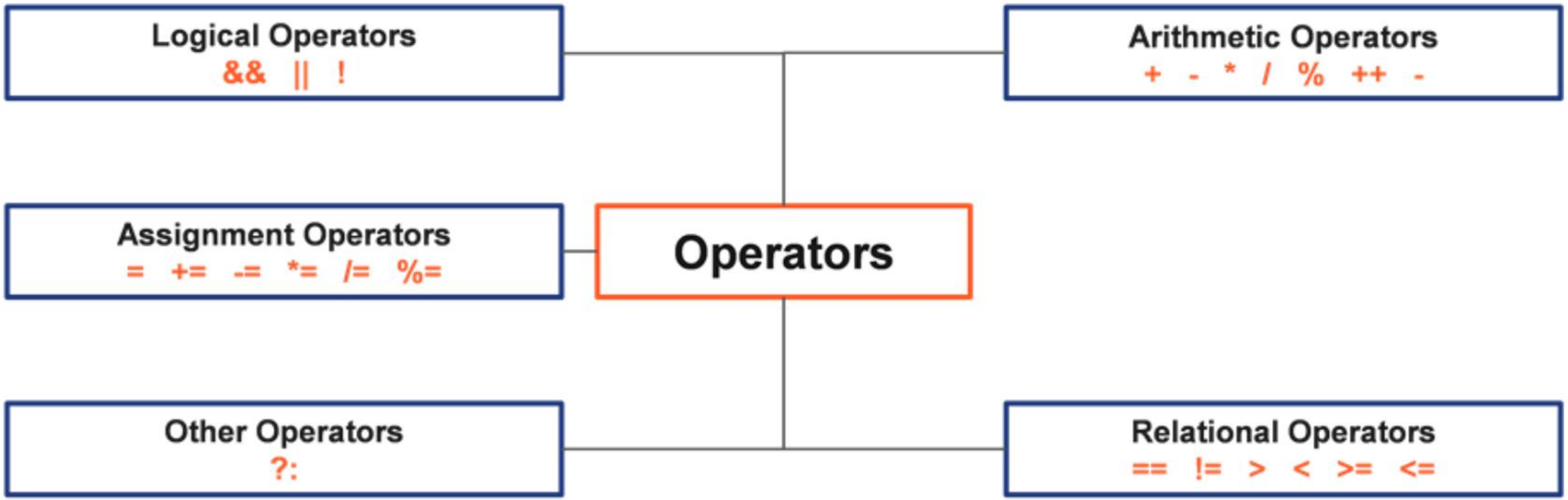
Java Operators

- Operations can be performed on the variables that hold data or on the literals itself.
- These operations can be arithmetic calculations, relational comparisons, etc.
- An operator is a special symbol used to perform operations on variables or literals called **operands**.
- The following figure shows an addition operator with two operands.



Different Types of Operators

- The given image lists the different types of Java operators.



Arithmetic Operators

- Arithmetic operators:
  - An arithmetic operator is a mathematical function that takes two operands and performs a calculation on them.
  - The operands can be literals or variables.
- Add two integer literals and store them in an integer variable called **sum.**

```
int sum = 20 + 30;
```

- Add two variables that hold integer values and store them in another integer variable called **sum.**

```
int number1 = 20;
int number2 = 40;
int sum = number1 + number2;
```

Arithmetic Operations on Literals

- A Java program can be used to multiply two integers and print the result on the console.
- The variable product holds the calculated value.
- The `System.out.println` statement is used to display the output on the console.

```
class Calculation {
    public static void main(String[] args) {
        int product = 42 * 23;
        System.out.println("The Product is ");
        System.out.println(product);
    }
}
```

Output

```
The Product is
966
```

```java
class Calculation {
    public static void main(String[] args) {
        int number1 = 42;
        int number2 = 23;
        int product = number1 * number2;
        System.out.println("The Product is ");
        System.out.println(product);
    }
}
```

Output

```
The Product is
966
```

Arithmetic Operations on Variables

- A Java program can be used to multiply two integers and print the result on the console.
- Two variables `number1` and `number2` hold the integer literals.
- The variable `product` holds the calculated value.
- The `System.out.println` statement is used to display the output on the console.

Calculation

Write a program to perform addition, subtraction, division, and multiplication on two numbers and display the result.

Declare and initialize values to the two numbers. The data type to be used and the value of the numbers are given in the table below.

Variable Name	Value
number1	50.79
number2	15

Note: In this example, the data types of the two variables are different.

DEMO

Relational Operators

- Relational operators:
  - Relational operators function by comparing two values and then by returning them true or false depending on the outcome of the comparison.
- Compare two integers to check the greatest of the two and store the result in a Boolean variable result.

```
boolean result = 20 > 30;
```

- Compare two variables that hold integer values to check the greatest of the two and store the result in a Boolean variable result.

```
int number1 = 20;
int number2 = 40;
boolean result = number1 > number2;
```

Relational Operations

- Compare the values of two integers and display true if the first number is greater than the second.
- The `boolean` variable `isGreatest` holds the comparison result of the two variables.

```
class Calculation {
    public static void main(String[] args) {
        int number1 = 42;
        int number2 = 23;
        boolean isGreatest = number1 > number2;
        System.out.println("Is number1 greater than number2 ?  ");
        System.out.println(isGreatest);
    }
}
```

Output

```
Is number1 greater than number2 ?
true
```

The `System.out.println` Statement

- To display the values of variables we can write the statement `'Is number 1> number2 ?'` in one `System.out.println` and the variable in another `System.out.println.`

```
System.out.println("Is number1 > number2 ? : ");
System.out.println(isGreatest);
```

- This can be simplified by using the `+` operator to concatenate or join the statement `'Is number 1> number2 ?'` which is a `String` and the variable `isGreatest` that is an integer. This will always evaluate to a `String` only.

```
System.out.println("Is number1 > number2 ? : " + isGreatest);
```

Greatest of Two Numbers

Write a program that will declare and initialize two numbers and display true, if the number1 is greater than the number2, or else display false.

Variable Name	Value
number1	56.78
number2	35

DEMO

Expressions

- An expression is a construct made up of variables, operators, etc., that are constructed according to the syntax of the language.
- In the below example `number1 * number2` is an expression.

```
class Calculation {
    public static void main(String[] args) {
        int number1 = 42;
        int number2 = 23;
        int product = number1 * number2;
        System.out.println("The Product is ");
        System.out.println(product);
    }
}
```

- Expressions always evaluate to a single value.

Quick Check

Find the expressions in the code below:

```java
class Calculation {
    public static void main(String[] args) {
        int number1 = 120;
        int number2 = 302;
        int sum = number1 + number2;
        System.out.println(sum);
        int difference = number2 - number1;
        System.out.println(difference);
    }
}
```

Quick Check: Solution

Find the expressions in the code below:

```
class Calculation {
    public static void main(String[] args) {
        int number1 = 120;
        int number2 = 302;
        int sum = number1 + number2;
        System.out.println(sum);
        int difference = number2 - number1;
        System.out.println(difference);
    }
}
```

Statements in Java

- Statements are equivalent to sentences in natural languages.
- In programming, a statement forms a complete line of execution.
- There are many types of statements:
  - Expression statements:
    - Expressions can be made into a statement by terminating the expression with a semicolon (;).
  - Declaration statements:
    - A declaration statement declares a variable.

Types of Statements

Expression Statements

```
int sum = number1 + number2;
```

```
int difference = number2 - number1;
```

```
boolean isGreatest = number1 > number2;
```

Declaration Statements

```
int number1 = 120;
```

```
boolean result = false;
```

```
float decimalNumber = 33.6f;
```