



Sukkur Institute of Business Administration University

Department of Computer Science

Programming Fundamentals

BS – I (CS/AI/SE)

Spring-2024

Lab # 01: Unleash the Java Giant: Building Your First Programs!

Instructor: Nimra Mughal

Objectives

After performing this lab, students will be able to

- What is JAVA?
- Features of Java.
- Installing JDK and setting path.
- Writing HelloWorld.java in Text Editor
- JAVA variables & data types.
- Input & Output
- Java Variable Type Conversion & Type Casting
- Lab tasks

What is JAVA?

Java: The Programming Powerhouse Behind Your Favorite Apps!

Imagine a language that can bring to life everything from sleek Android phones to powerful web servers. That's the magic of Java, a programming language that's been around **since 1995 and is still going strong!**

Java isn't just a language, it's a complete **platform**. Think of it as a giant toolbox filled with building blocks for creating all sorts of awesome things. You can use Java to:

- **Build mobile apps:** From the games you play to the banking apps you use, many of them are powered by Java!
- **Develop websites:** Java helps create the backbone of many websites, making them interactive and dynamic.
- **Craft desktop software:** From productivity tools to creative software, Java can be used to build programs for your computer.
- **Power the internet:** Java is used in servers that keep the internet running smoothly, behind the scenes!

There are many java versions that has been released.

A timeline of Java versions: <https://www.codejava.net/java-se/java-se-versions-history>

Download Java for yourself: <https://www.oracle.com/java/technologies/downloads/>

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.

Java Platforms

According to Oracle, there are four platforms of the Java programming language

- Java Platform, Standard Edition (**Java SE**)
- Java Platform, Enterprise Edition (**Java EE**)
- Java Platform, Micro Edition (**Java ME**)
- JavaFX/OpenJFX

Ready to dive into the world of Java? There are tons of resources available online and in libraries to help you get started. Remember, the journey of a thousand programs begins with a single line of code!

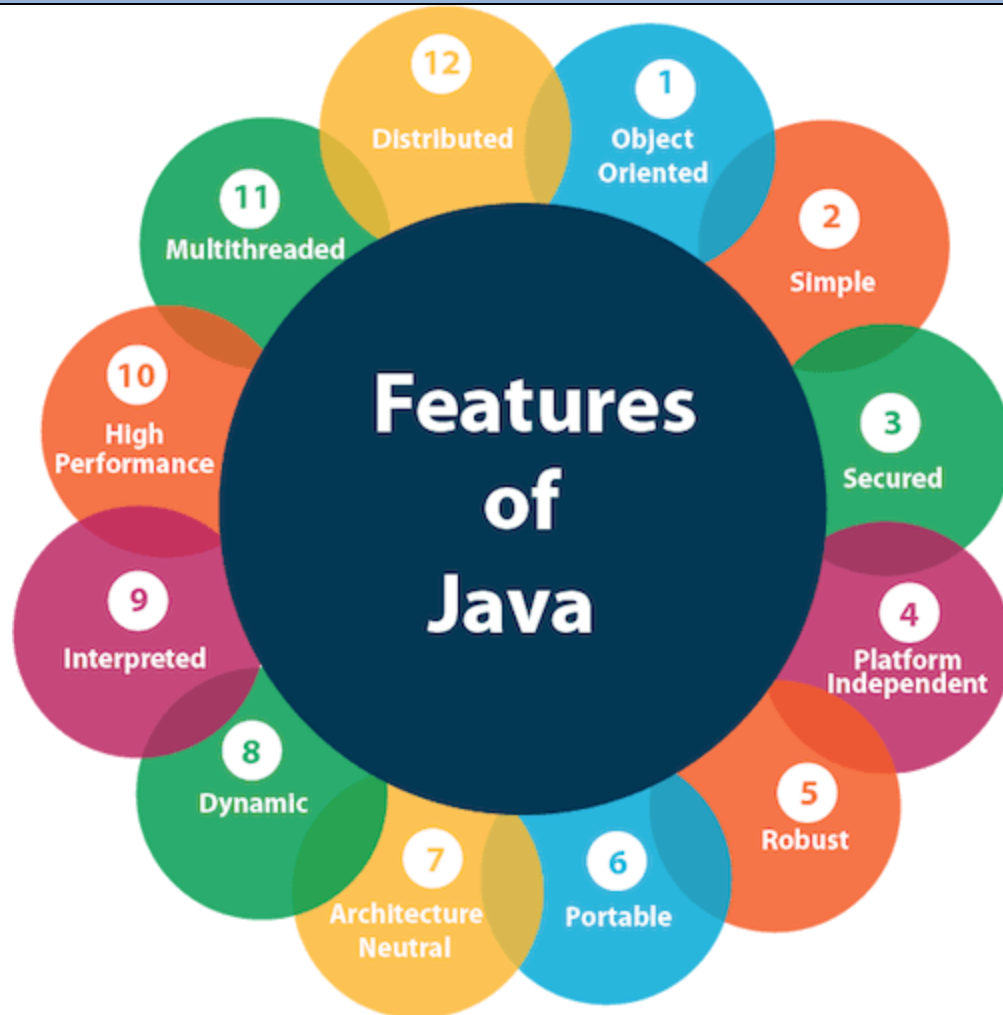
Bonus tip: Check out these websites for fun Java tutorials and projects:

Codecademy: <https://www.codecademy.com/learn/learn-java>

Udacity: <https://www.udacity.com/course/java-programming-nanodegree--nd079>

Coursera: <https://www.coursera.org/specializations/object-oriented-programming>

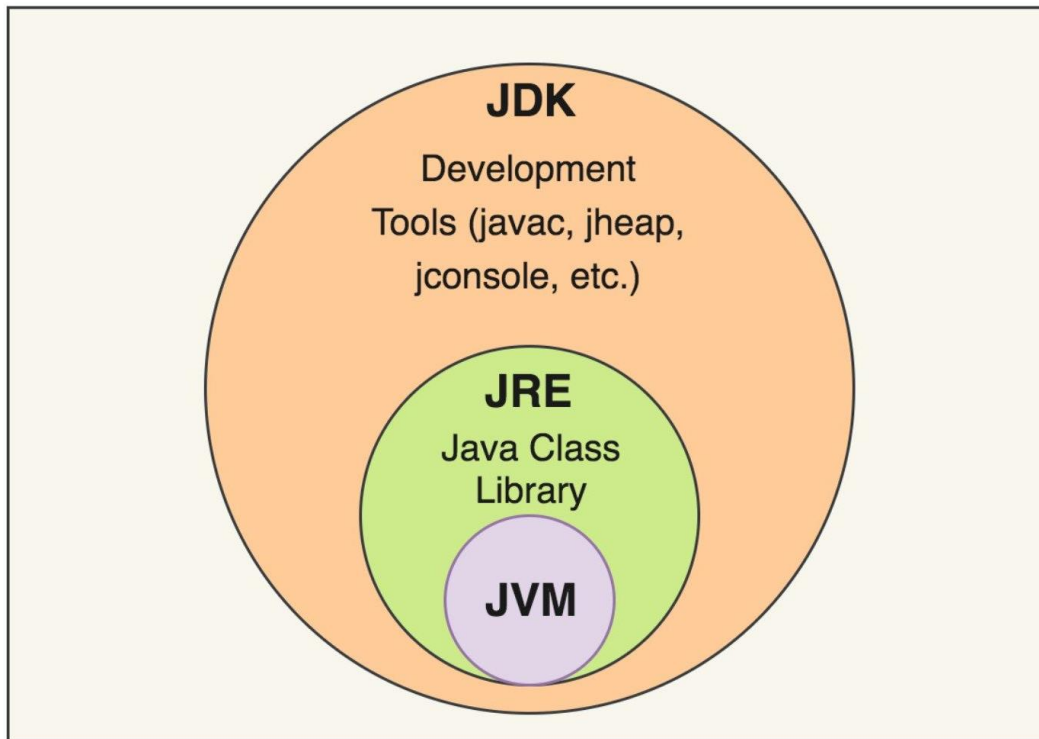
Features of JAVA



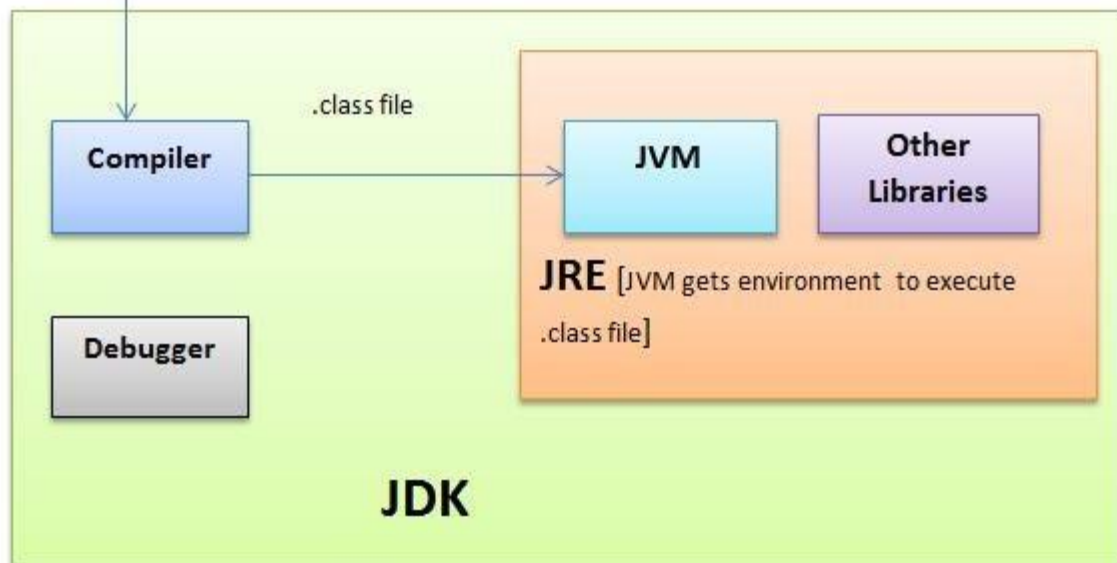
By making it the system's responsibility to do this final conversion, Java has created a write once, run anywhere language where anyone can hand you a Java program and you can run it on your machine

"Write once, run everywhere" WORA

JAVA Basics



Source files (.java files)



JDK

- It stands for **Java Development Kit**, is a software development environment used for developing Java applications and applets.
- It compiles and executes new and already built applications.
- It is a collection of development tools as well as Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

JRE

- It stands for **Java Runtime Environment**. The Java Runtime Environment provides the minimum requirements for executing a Java application.
- It consists of the **Java Virtual Machine (JVM)**, **interpreter**, **JIT**, **core classes**, and supporting files.

JVM

- It stands for **Java Virtual Machine (JVM)**
- It is responsible for executing bytecode where interpreter provides machine code for the current machine and has JIT as well.

JIT

- It stands for **Just-in-time** Compiler, is the part of the JVM that is used to speed up the execution time.
- JIT interpret parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for full interpretation.

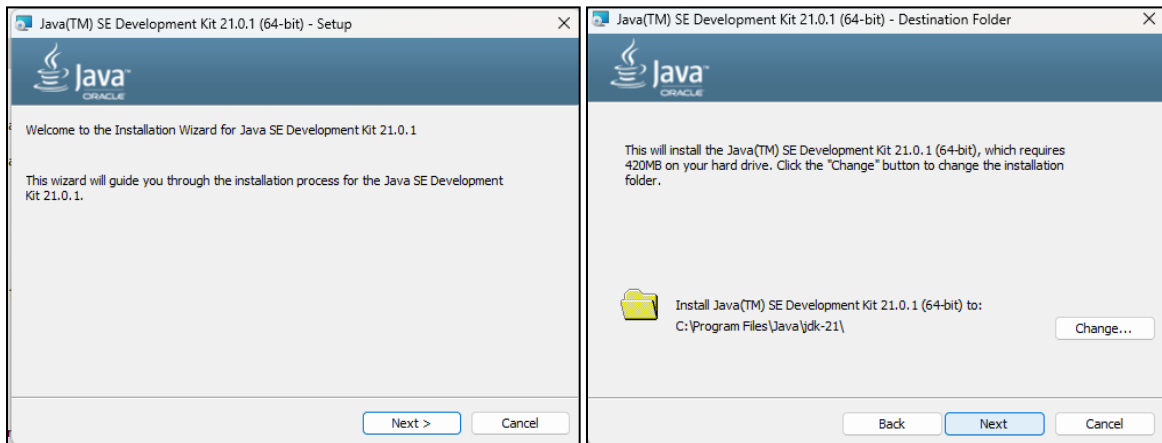
Installing JDK and setting path

To develop Java applications on our computers, we require a JDK. Visit the link below to download the JDK setup.

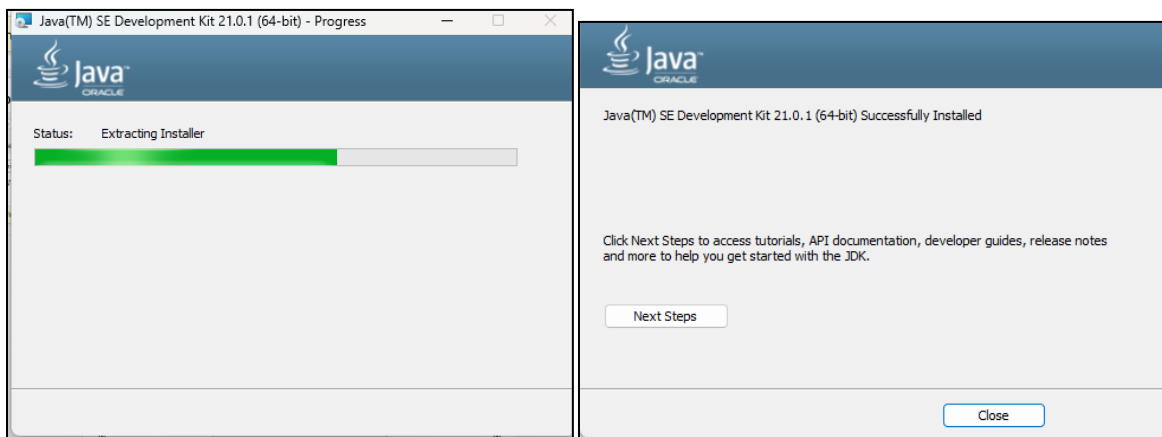
https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe

Double click downloaded JDK and click YES to allow JDK to make changes to system and then click NEXT.

Installation location by default is C:\Program Files\Java\jdk-21\



Installation in progress



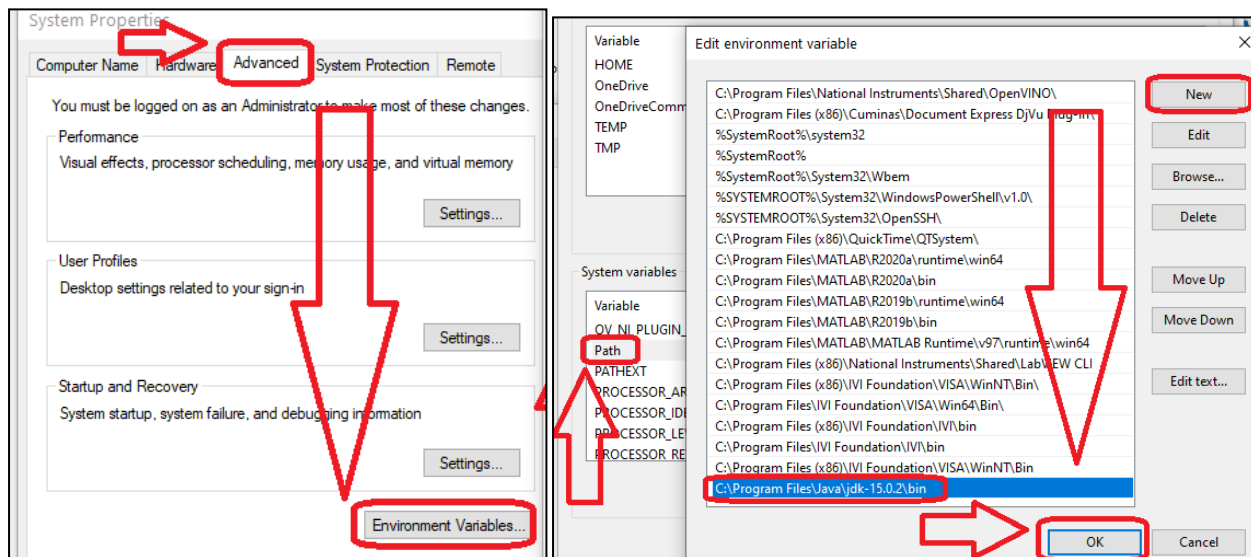
JDK successfully installed, click **CLOSE**

We need to set Java path so that whenever java program will be compiled or executed it can locate JDK easily. Path can be set in two ways i.e. temporarily and permanently.

To set java path temporarily, open command prompt and run command “**set path=C:\Program Files\Java\jdk-21\bin**”. Once the command prompt is closed, java path needs to be set again.

To set java path permanently, this path “C:\Program Files\Java\jdk-21 \bin” should be added as per following steps.

Click the windows button and Type “**Environmental variables**” Then follow the steps shown in screenshots



To check if the Java compiler is installed, **open command prompt** and run “javac -version”.

If the command returns something such as “javac 21.0.1”, it means JDK having Java compiler is installed successfully.

To check version information of JDK, JRE and JVM, open command prompt and run command “java -version”.

```

C:\Users\nimra>java -version
java version "21.0.2" 2024-01-16 LTS
Java(TM) SE Runtime Environment (build 21.0.2+13-LTS-58)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.2+13-LTS-58, mixed mode, sharing)


C:\Users\nimra>javac -version
javac 21.0.2

C:\Users\nimra>
  
```

Writing HelloWorld.java in Text Editor

Follow the below given steps:

- i. Run notepad and enter below given code. Save this file with Class name and end it with “.java” extension. Save it on any location!

 hello - Notepad

File Edit Format View Help


```
class Hello{
    public static void main(String args[])
    {

        System.out.println("Hello to Java");

    }
}
```

- ii. Go to search bar in taskbar, write cmd and Open Command Prompt then write the following commands;
 - a. `cd E:\Collaborative Learning\Spring_24_BS-II_OOP\Codes`
(//Here you will type the path of your Hello.java program)
 - b. `javac HelloWorld.java`
 - c. `java HelloWorld`

The Java programming language compiler (javac) takes your source file and translates the code into instructions known as bytecodes. “Java ClassName” will enable Java virtual machine to run your application/code.

 Command Prompt

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nimra>E:

E:\>cd "E:\Collaborative Learning\Spring_24_BS-II_OOP\Codes"

E:\Collaborative Learning\Spring_24_BS-II_OOP\Codes>javac hello.java

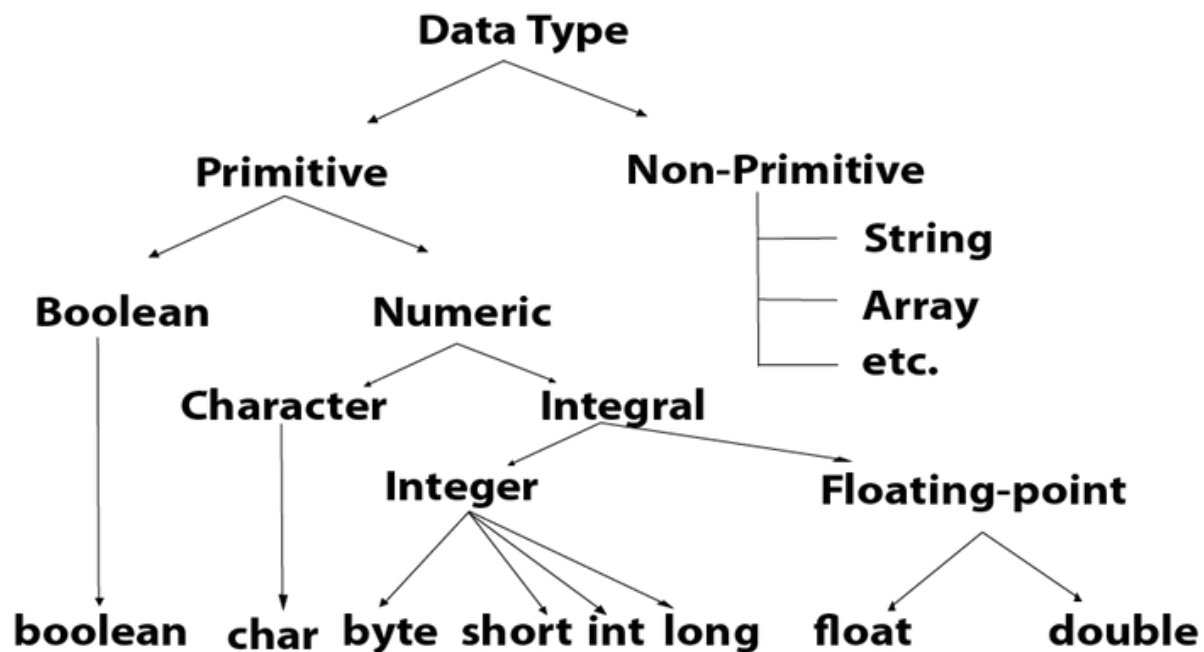
E:\Collaborative Learning\Spring_24_BS-II_OOP\Codes>java hello
```


Variables and data types in JAVA

In Java, there are three types of variables:

- Local Variables
- Instance Variables
- Static Variables

Data Types in Java



Data Type	Default Value	Default size
Boolean	false	1 bit
Char	'\u0000'	2 byte
Byte	0	1 byte
Short	0	2 byte
Int	0	4 byte
Long	0L	8 byte
Float	0.0f	4 byte
Double	0.0d	8 byte

Example: Types of Variables in Java

```

class Variab {
    int InsVarExam = 29; //instance variable
    static int IsStatVar = 15; //static variable
    void method() {
        int IsLocalVar = 90; //local variable
    }
}

```

Input & Output**Output in Java Syntax:**

```
System.out.println("Hello World");
```

Input in Java Syntax:

```

// import library
import java.util.Scanner;
class Demo{
    public static void main(String args[])
    {
        // Creating scanner object
        Scanner ip = new Scanner(System.in); //system.in represents that the input is given via keyboard
        //Taking input from user
        int ipInt = ip.nextInt(); // Integer Input
        double ipDb = ip.nextDouble(); // Double Input
        System.out.println(ipInt);
        System.out.println(ipDb);
    }
}

```

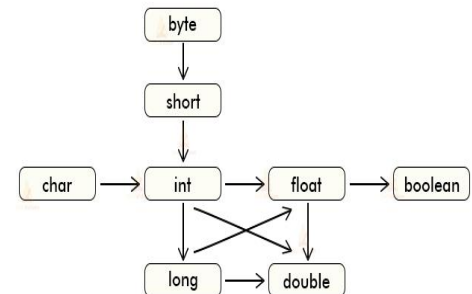
Java Variable Type Conversion & Type Casting

A variable of one type can receive the value of another type.
Here there are 2 cases.

Case 1: A variable of smaller capacity is assigned to another variable of bigger capacity.

```
double d ;
int i = 10;
d = i;
```

Implicit Type Conversion in Java



This process is Automatic, and non-explicit is known as **type conversion**

Case 2: Variable of larger capacity is be assigned to another variable of smaller capacity.

```
double d = 10;
int i;
i = (int) d
```

Type Cast Operator

In such cases, you have to explicitly specify the type cast operator. This process is known as **type casting**.

In case, you do not specify a type cast operator; the compiler gives an error. Since this rule is enforced by the compiler, it makes the programmer aware that the conversion he is about to do may cause some loss in data and prevents accidental losses.

Example: To Understand Type Casting

```
class Demo {
    public static void main(String args[]) {
        byte x;
        int a = 270;
        double b = 128.128;
        System.out.println("int converted to byte");
        x = (byte) a;
        System.out.println("a and x " + a + " " + x);
    }
}
```

```

    System.out.println("double converted to int");
    a = (int) b;
    System.out.println("b and a " + b + " " + a);
    System.out.println("\ndouble converted to byte");
    x = (byte)b;
    System.out.println("b and x " + b + " " + x);
}
}

```

Output:

int converted to byte

a and x 270 14

double converted to int

b and a 128.128 128

double converted to byte

b and x 128.128 -128

Exercises

Question 1: (JAVA Environment Installation & Error Messages)

1. Set up your Java development environment:
2. Explore error identification and correction:
 - Carefully examine the following code snippets, paying close attention to syntax and common errors:

// Snippet 1

```
System.out.printn("Hello World");
```

// Snippet 2

```
System.out.printn(Hello World);
```

// Snippet 3

```
System.out.println"Hello World";
```

// Snippet 4

println("Hello World);

- Before compiling, try to identify the errors in each snippet.
 - Record your predictions about the errors and their causes.
3. Experiment with compilation and error messages:
- Compile each snippet individually.
 - Observe the error messages generated by the compiler.
 - Compare the actual error messages with your predictions.
 - Analyze each error message to understand its meaning and pinpoint the specific issue.
4. Fix the errors:
- Correct each code snippet based on the error messages and your understanding of Java syntax.
 - Recompile the corrected code to ensure it runs without errors.
5. Reflect and discuss:
- Discuss the types of errors you encountered and their common causes.
 - Share strategies for effectively identifying and correcting errors in Java code.
 - Reflect on the importance of careful coding practices and attention to detail.

Question: 2 (Input)

Create a program that asks the user for their name, age, height (in meters), and favorite color.

Extension: can you convert the height in cm?

Question: 3 (Basic Arithmetic Operation)

- Challenge: Explore different numerical data types through calculations.
- Requirements:
 - Declare variables of type int, double, and float.
 - Assign various values to them.
 - Perform simple arithmetic operations (addition, subtraction, multiplication, division).
 - Print the results, carefully observing the output format for each data type.

Question: 4 (Type casting)

Perform division using two int variables and store the result in float variable and print the results.

Question: 5 Can you cast string into int?

Question: 6 Why JAVA when there are other OOP languages?