

# Advanced Programming

## Lecture 1

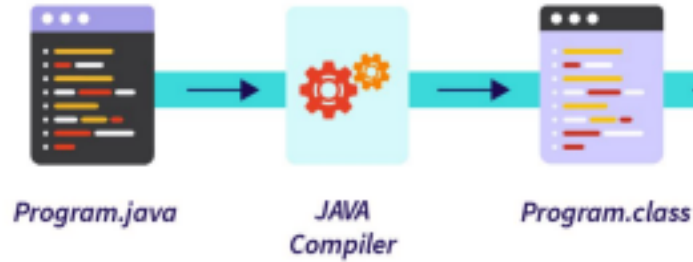
Md. Tariquzzaman (TZF)  
Lecturer, BUBT

### How does Java Work?



SCALER  
Topics

# How does Java Work?



## Java Compiler

- **Java Source Code:** Human-readable code written by developers in `.java` files.
- **Compilation:** The Java compiler (`javac`) takes the source code and compiles it into bytecode, which is stored in `.class` files.
- **Bytecode:** A platform-independent, intermediate representation of the code. It is not tied to any specific machine architecture.

3

## How does Java Work?

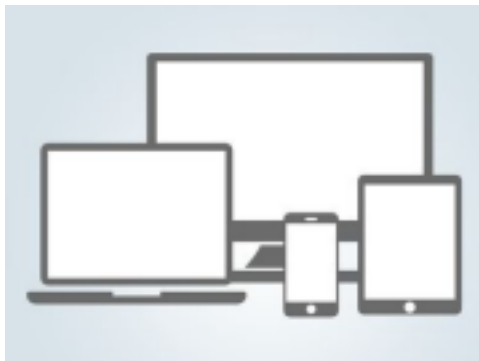


## Java Virtual Machine (JVM)

- **Class Loader:** Loads `.class` files containing bytecode into memory.
- **Bytecode Verifier:** Ensures the bytecode is valid and does not violate Java's security constraints.
- **Interpreter:** Reads and executes bytecode instructions. JVM translates bytecode into machine code specific to the host machine.
- **Just-In-Time (JIT) Compiler:** During execution, the JVM can optimize performance by compiling frequently executed bytecode into native machine code.

4

# How is Java a cross-platform language?



- **Platform Independence:**
  - Java programs are written in a standardized language and compiled into bytecode.
  - The same `.class` files can run on any device or operating system that has a compatible JVM.
- **JVM Implementation:** JVMs are available for different operating systems and hardware architectures. This allows Java bytecode to be executed on any platform with a JVM, making Java programs highly portable.

# Your First Code in Java

```
// Your First Program

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

In C: “int main()”

In Java: “class <filename> {  
public static void main(String[] args) { .....  
}  
}”

In C: “printf”

In Java: “System.out.println” <sup>6</sup>

## Your First Code in Java

class file on the command line

```
java HelloWorld
```

```
// Your First Program

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```



```
name of the source file on the command line  
javac HelloWorld.java
```

**public:** So that JVM can execute the method from anywhere.

**static:** The main method is to be called without an object. The modifiers are public and static can be written in either order.

**void:** The main method doesn't return anything.

**main():** Name configured in the JVM. The main method must be inside the class definition. The compiler executes the codes starting always from the main function.

**String[]:** The main method accepts a single argument, i.e., an array of elements of type String.

## Input Output in Java

```
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);

        // Getting float input
        System.out.print("Enter float: ");
        float myFloat = input.nextFloat();
        System.out.println("Float entered = " + myFloat);

        // Getting double input
        System.out.print("Enter double: ");
        double myDouble = input.nextDouble();
        System.out.println("Double entered = " + myDouble);

        // Getting String input
        System.out.print("Enter text: ");
        String myString = input.next();
        System.out.println("Text entered = " + myString);
    }
}
```

# Why do we need OOP?



# Why do we need OOP?



A class is a blueprint or template for creating objects. Each Pokemon species has a set of attributes (like type, abilities, and moves) and methods (like attacks).

When you catch a Pikachu, you create an object of the Pikachu class with specific values for its attributes.



# Why do we need OOP?



Note: These are just examples. Don't write these on exam as definitions, use programming definition

# Data Types

```
int = 5; // Integer (whole number)
```

```
float =
```

```
5.99f; // Floating point number (32 bit)
```

```
double =
```

```
19.99; // Double number (64 bit)
```

```
char = 'D'; //
```

```
Character
```

```
boolean = true; // Boolean
```

```
String = "Hello"; // String
```

# Operators





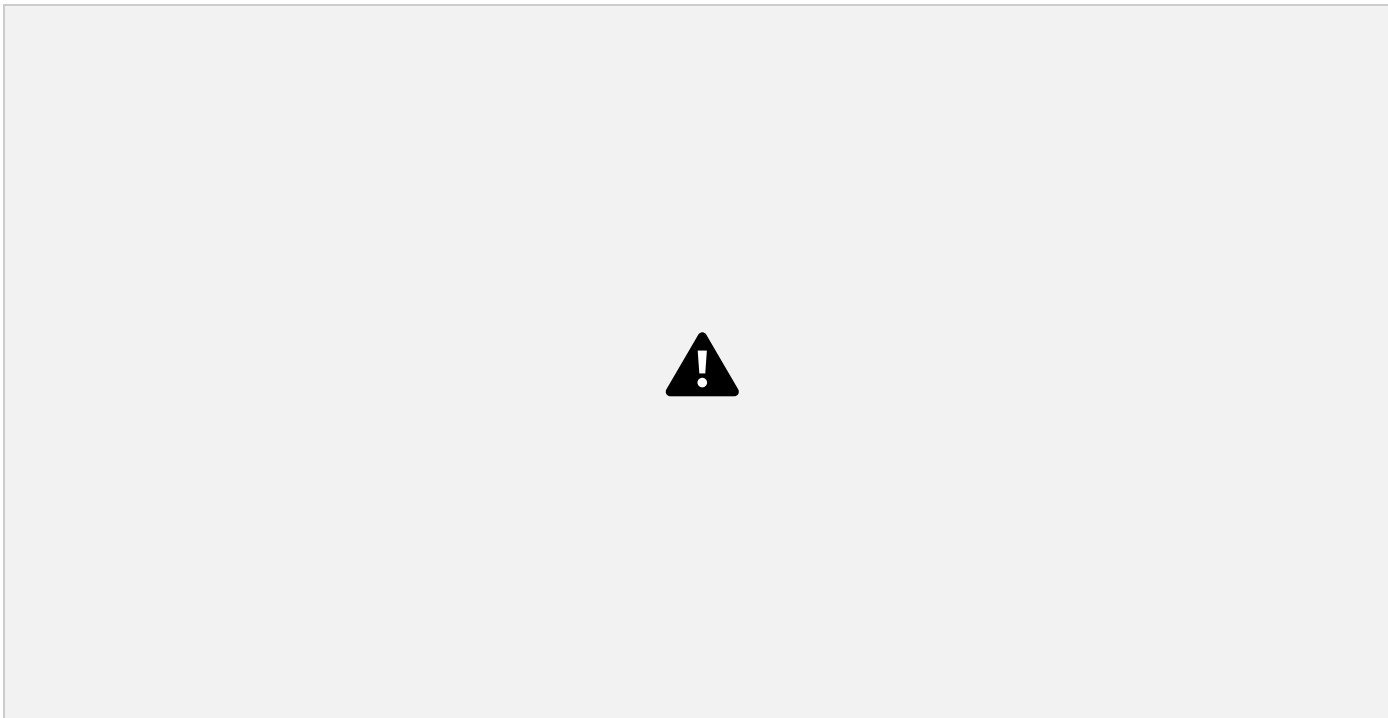
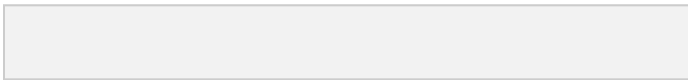
# Operators





14

# Operators



# Operators



this website: [Java Operators](#) 16

Test Your Skills in

# Conditionals/Control Flow



Test Your Skills in this website: [Java The else if Statement](#) 17

# Loops



Test Your Skills in this website: [Java While Loop](#), [Java For Loop](#)

# Arrays



Test Your Skills in this website: [Java Arrays](#)

# That's All!