FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Java in Web Programming

Elena Atanasoska

What is Java?

- General-purpose, object-oriented programming language.
- Widely used for web, mobile, and enterprise applications.

Basic Java Syntax

- Case-sensitive, uses curly braces {} to define code blocks.
- Every statement ends with a semicolon ; .
- Entry point: public static void main(String[] args).

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

Variables and Data Types

- Primitive types: int, double, char, boolean, etc.
- Object types: classes like String, arrays, and collections.
- Declaration: type variableName = value;.

```
int number = 5;
String text = "Spring Boot";
boolean isActive = true;
```

Control Structures

- if-else, switch for decision making.
- Loops: for, while, do-while.

```
if (number > 0) {
    System.out.println("Positive number");
} else {
    System.out.println("Non-positive number");
}
```

```
for (int i = 0; i < 5; i++) {
    System.out.println(i);
}
```

Methods (Functions)

- Reusable blocks of code.
- Defined with return type, name, parameters.
- Can return values or be void.

```
public int addNumbers(int a, int b) {
    return a + b;
}
```

Object-Oriented Programming - Classes and Objects

- Java is object-oriented. Code is structured as classes with fields and methods.
- Classes are blueprints for creating objects (instances).

```java
public class Car {
    String model;
    int year;

    public Car(String model, int year) {
        this.model = model;
        this.year = year;
    }

    public void startEngine() {
        System.out.println("Engine started");
    }
}
```

```java
Car myCar = new Car("Toyota", 2022);

myCar.startEngine();
```

Inheritance and Interfaces

- Inheritance allows one class to inherit fields and methods from another (extends).
- Interfaces define methods that must be implemented (implements).

```
interface Drivable {
    void drive();
}
public class Vehicle implements Drivable{
    public void start() {
        System.out.println("Vehicle started");
    }

    @Override
    public void drive() {
        System.out.println("Vehicle is
         driving");
    }
}
```

```
public class Car extends Vehicle {
    public void drive() {
        System.out.println("Car is
         driving");
    }
}
```

Error and Exception Handling

- Use try-catch to handle exceptions and avoid program crashes.
- throw exceptions when necessary.

```
try {
    int result = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Error: Division
by zero");
}
```

Collections Framework

- Store and manage groups of objects: List, Set, Map.
- Common methods: add(), remove(), get().

```
List<String> names = new ArrayList<>();
names.add("John");
names.add("Jane");

for (String name : names) {
    System.out.println(name);
}
```

Online Resources for Learning Java

- Official Documentation
  - https://docs.oracle.com/javase/tutorial/java/index.html

- Tutorials and Guides
  - W3Schools Java Tutorial - https://www.w3schools.com/java/
  - Learn Java on Codecademy – https://www.codecademy.com/learn/learn-java