Digital Content on Java programming

Subject Name: Java Programming

Subject Code: 28541

Md. Sabuj Rana
Instructor(Tech/Computer)
Computer Science and Technology
Khulna Polytechnic Institute

Presentation: Introduction to Basic Java Programming

What is Java?

- **Definition**: Java is a high-level, object-oriented programming language.
- **Features**:
  - Platform-independent (Write Once, Run Anywhere)
  - Robust and secure
  - Extensive libraries and frameworks
  - Versatile: Web development, mobile apps, backend systems, and more

---

Why Learn Java?

- **Popular**: Widely used in enterprise applications.
- **Versatile**: Suitable for various domains like web, mobile, and IoT.
- **High Demand**: Java developers are highly sought after.
- **Community Support**: Large and active community for help and resources.

Installing Java

- **Steps to Install Java**:
  1. Visit the official Oracle website: https://oracle.com/java.
  2. Download the latest JDK (Java Development Kit) for your operating system.
  3. Install Java and set the `JAVA_HOME` environment variable.
- **Tools to Use**:
  - IDEs: IntelliJ IDEA, Eclipse, or NetBeans.

Java Basics

- **Hello World Example**:
- `public class HelloWorld {`

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

- **Variables**:

```
int age = 25;
String name = "Alice";
```

- **Data Types**:
  - Primitive: int, double, char, boolean
  - Non-Primitive: Strings, Arrays, Objects

Control Structures

- **Conditionals**:

```
if (age > 18) {
    System.out.println("Adult");
} else {
    System.out.println("Minor");
}
```

- **Loops**:
  - **For Loop**:

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

  - **While Loop**:

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

Methods

- **Definition**: Reusable blocks of code.
- **Example**:

```
public class Greetings {
    public static String greet(String name) {
        return "Hello, " + name + "!";
    }

    public static void main(String[] args) {
        System.out.println(greet("Alice"));
    }
}
```

Object-Oriented Programming (OOP)

- **Core Concepts**:
  - Classes and Objects
  - Inheritance
  - Polymorphism
  - Encapsulation
  - Abstraction
- **Example**:

```
class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal myDog = new Dog();
        myDog.sound();
    }
}
```

Working with Libraries

- **What are Libraries?**
  - Pre-written code that you can use in your programs.
- **Popular Libraries**:
  - Apache Commons (Utilities)
  - Google Guava (Collections and Utilities)
  - Jackson (JSON Processing)
- **Example**:

```
import java.util.ArrayList;

public class Example {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        System.out.println(list);
    }
```

- `}`

Debugging Tips

- **Common Errors**:
  - Syntax errors
  - Runtime errors
  - Logical errors
- **Tips**:
  - Read error messages carefully.
  - Use debugging tools in your IDE.
  - Write and test code in small parts.

Practice Makes Perfect

- **Resources**:
  - Online Platforms: Codecademy, LeetCode, HackerRank.
  - Java Documentation: https://docs.oracle.com/javase.
- **Projects to Try**:
  - Build a calculator.
  - Create a simple game (e.g., Tic-Tac-Toe).
  - Develop a to-do list app.

Q&A

- **Ask Me Anything!**
- Contact Information: [Your Email/LinkedIn Profile]

Thank You!

- **Message**: Thank you for attending!
- **Closing Note**: Happy Coding!