



Summary of knowledge prior Midterm

University American College Skopje
School of Computer Science and Information Technology
Course: Programming Languages
Prepared by: Dejan Mitov

Table of Contents


1. Introduction to Java
2. Expressions, Control Structures, Loops, and Arrays -
3. Classes, Arrays of Reference Type Objects, Command Prompt Input
4. Inheritance and Abstract Classes
5. Overloading, Java API, Advanced Structures, Nesting, Wrapping2.
6. Key Concepts
7. Terminology
8. Practice Questions
9. Summary and Key Takeaways



Expanded Key
Summarization of
Each Lecture

Introduction to Java

- **What is Java?:** An introduction to the Java programming language, its history, and how it compares to other languages.
- **Setting Up the Environment:** Step-by-step guide on installing the Java Development Kit (JDK) and setting up the environment variables.
- **Java Program Structure:** Overview of how a basic Java program is structured, focusing on the `main` method as the entry point.
- **Compilation and Execution:** Explanation of how to compile a Java program using `javac` and run it using `java`.
- **Hello, World Example:** A simple "Hello, World" program is dissected to understand the basic elements of a Java program.



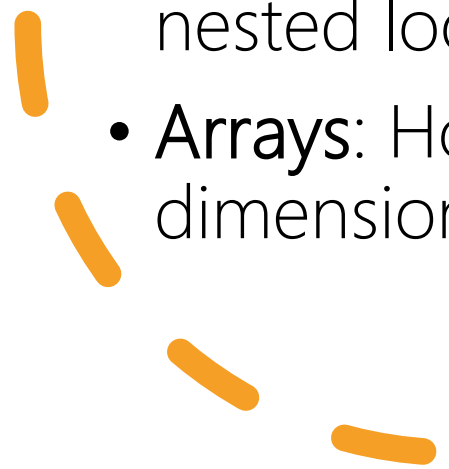
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

}





Expressions, Control Structures, Loops, and Arrays

- **Expressions:** Discussion on arithmetic and logical expressions, and their evaluation.
 - **Control Structures:** Detailed explanation of `if`, `else`, and `switch` statements with examples.
 - **Loops:** Introduction to `for`, `while`, and `do-while` loops, including nested loops.
 - **Arrays:** How to declare, initialize, and manipulate single and multi-dimensional arrays.
- 

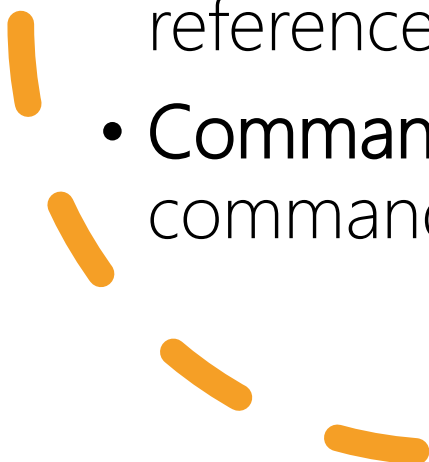
```
if (x > y) {  
    System.out.println("x is greater than y");  
} else {  
    System.out.println("y is greater than x");  
}
```

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

```
int[] numbers = {1, 2, 3, 4, 5};
```



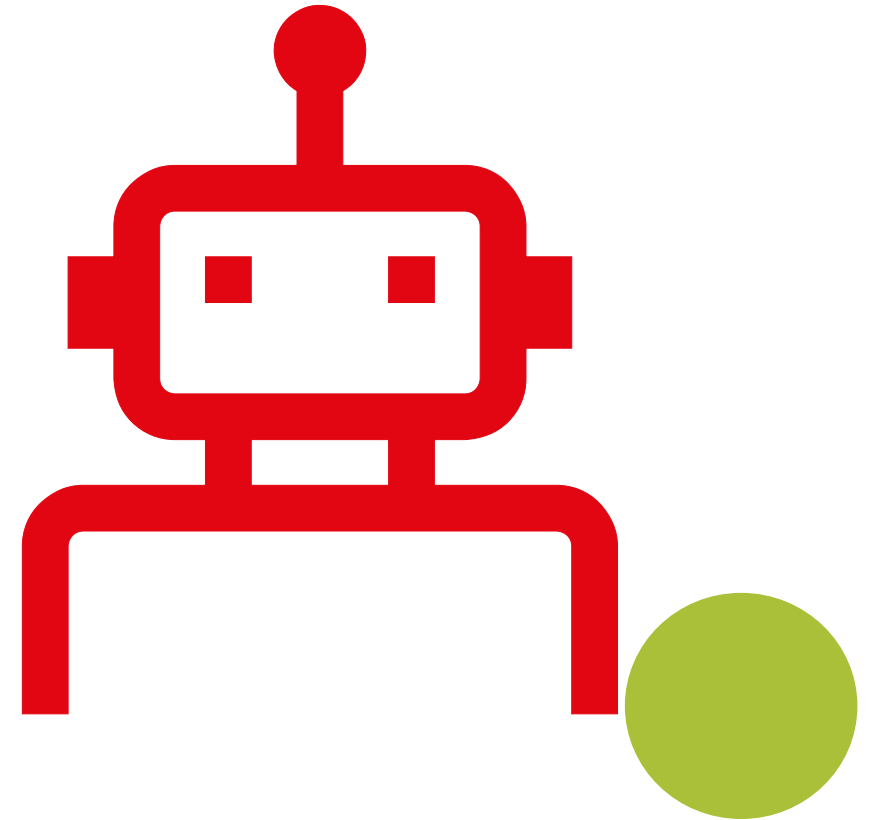
Classes, Arrays of Reference Type Objects, Command Prompt Input

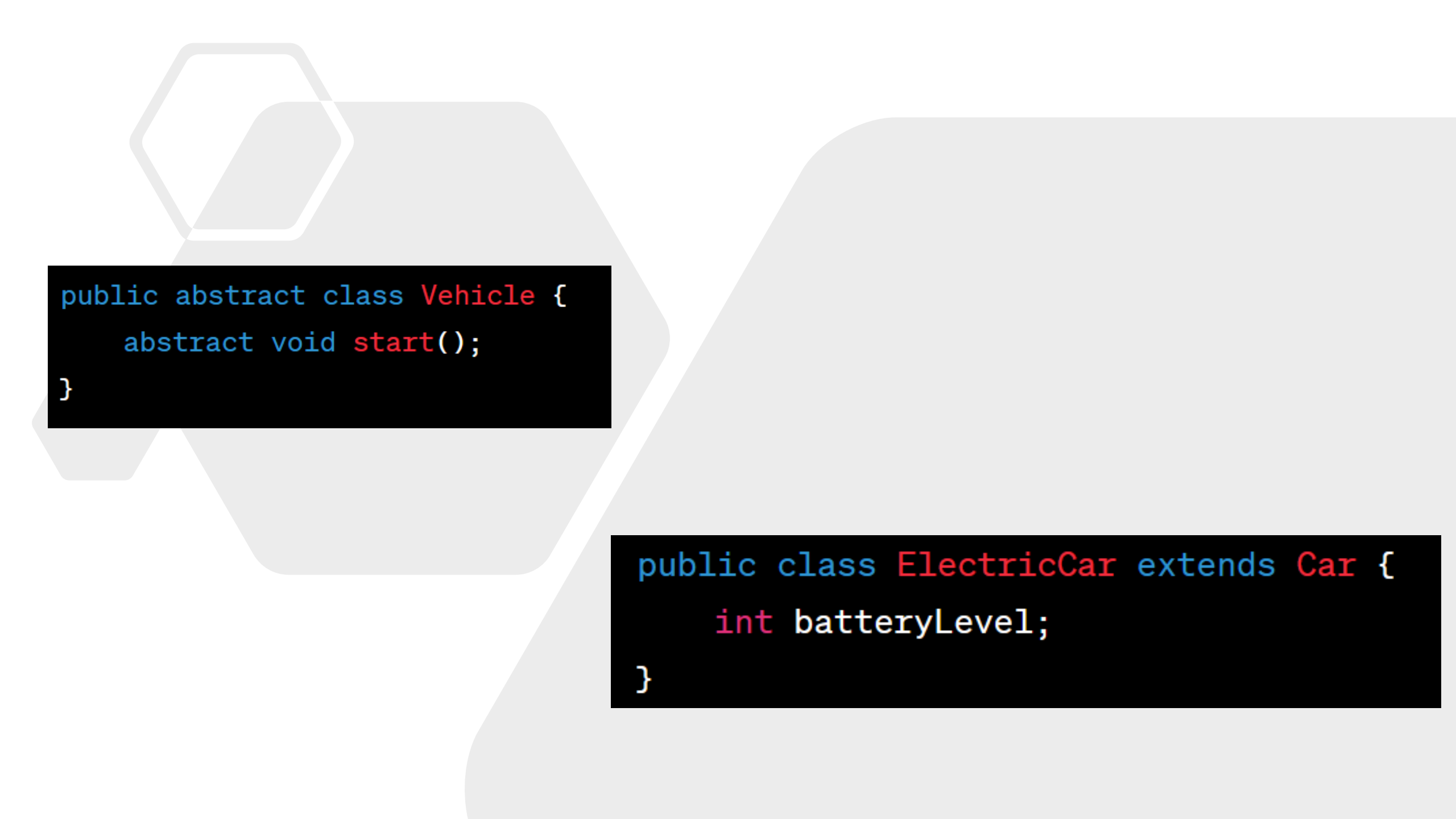
- **Object-Oriented Programming:** Introduction to classes and objects as the cornerstone of OOP in Java.
 - **Creating Classes and Objects:** Step-by-step guide on defining classes, creating objects, and invoking methods.
 - **Arrays of Reference Types:** How to create and use arrays that hold references to objects.
 - **Command Prompt Input:** Techniques for reading input from the command prompt using `Scanner` and `BufferedReader`.
- 


```
public class Car {  
    String make;  
    String model;  
    public Car(String make, String model) {  
        this.make = make;  
        this.model = model;  
    }  
}
```

Inheritance and Abstract Classes


- **Inheritance Basics:** Explanation of how inheritance allows a class to use attributes and methods from another class.
- **Types of Inheritance:** Single, multiple, multilevel, and hierarchical inheritance are discussed.
- **Abstract Classes and Methods:** Definition and usage of abstract classes and methods to achieve abstraction.





```
public abstract class Vehicle {  
    abstract void start();  
}
```

```
public class ElectricCar extends Car {  
    int batteryLevel;  
}
```

- 
- **Method Overloading:** How and why to overload methods within a class.
 - **Java API:** An overview of useful classes and methods available in the Java API.
 - **Advanced Structures:** Discussion of data structures like LinkedList, HashMap, and TreeSet.
 - **Nesting and Wrapping:** Advanced OOP techniques involving nested classes and object composition.



Overloading,
Java API,
Advanced
Structures,
Nesting,
Wrapping

```
import java.util.HashMap;  
HashMap<String, Integer> map = new HashMap<>();
```

```
public void print(int x) { /*...*/ }  
public void print(double x) { /*...*/ }
```

Key Concepts

- Basic Structure of Java Programs
- Control Structures (`if`, `else`, `switch`)
- Loops (`for`, `while`, `do-while`)
- Arrays and Multi-dimensional Arrays
- Object-Oriented Programming (OOP)
- Inheritance and Abstract Classes- Method Overloading- Advanced Data Structures

Terminology

- **Main Method:** The entry point of a Java application
- **Inheritance:** A mechanism where a new class is based on an existing class.
- **Abstract Class:** A class that cannot be instantiated and is meant to be subclassed.
- **Method Overloading:** Having multiple methods in the same class with the same name but different parameters.





Practice Questions

1. What is the purpose of the ``main`` method in a Java program?
2. How do you declare a multi-dimensional array in Java?
3. Explain the concept of inheritance with an example.
4. What is method overloading? Provide an example.
5. Describe how to take input from the command prompt in Java.



Summary and Key Takeaways

- Java is a robust and versatile programming language that supports object-oriented programming.
 - Understanding control structures, loops, and arrays is foundational for any Java developer.
 - Concepts like inheritance, method overloading, and advanced data structures enable you to write more efficient, maintainable code.
- 