Robotics Study Guide

This guide covers the key concepts from the first three chapters of your textbook to help you prepare for your exam.

Chapter 1: World of Robotics

This chapter introduces the foundational concepts of robotics and safety.

- **What is a Robot?**
  - A robot is a machine that can perform three key actions: **SENSE** (detect things around it), **THINK** (make decisions with a computer brain), and **ACT** (move and perform tasks).
  - Unlike a simple machine like a washing machine, a robot can adapt to situations and make its own decisions.
  - Real-world examples include the Roomba vacuum, factory robots, and delivery drones.
- **Types of Robots Around Us**
  - You should be able to recognize different categories of robots, including:
    - **Home Helper Robots** (e.g., Alexa, robot lawn mowers).
    - **Industrial Robots** (e.g., assembly line, packaging robots).
    - **Medical Helper Robots** (e.g., surgical, therapy robots).
    - **Transportation Robots** (e.g., self-driving cars, space robots).
    - **Humanoid Robots** (e.g., ASIMO, Sophia).

Chapter 2: Introduction to Programming

This chapter covers the basic theory of programming and how computers think.

- **What is Programming?**
  1. Programming is the process of writing very specific, step-by-step instructions for a computer to follow.
  2. Computers are literal and do exactly what they are told—nothing more, nothing less.
  3. A good analogy is a recipe: the program is the recipe, the input is the ingredients, and the output is the final dish.
- **How Computers Think**
  1. Computers process instructions **sequentially**, meaning they follow them in order (1, 2, 3...) and cannot skip ahead.
  2. They use **binary thinking**, where everything is either 0 or 1 (ON/OFF, TRUE/FALSE). Computers cannot understand uncertainty like "maybe".
- **Introduction to Algorithms**

1. An algorithm is a set of step-by-step instructions for solving a problem or completing a task.
2. An algorithm is the *plan* (like a recipe), while a **program** is that plan written in a specific programming language.

- **The 3 Building Blocks of Programming**
  1. **Sequence:** Instructions are executed one after another, in a specific order. Changing the order changes the result.
  2. **Loops (Repetition):** Used to repeat the same actions multiple times without rewriting the code.
  3. **Conditions (Decisions):** Allows a program to make choices using "IF-THEN" logic. The program checks if a condition is true or false and acts accordingly.

Chapter 3: Programming Fundamentals

This chapter dives into the practical tools and concepts for writing code.

- **The IDE (Integrated Development Environment)**
  - An IDE is a software application that provides a complete workspace for programming, like a digital workshop.
  - Key components include a **Code Editor** (to type code), a **Compiler** (to translate code for the computer), and a **Debugger** (to find errors).
  - Features like **syntax highlighting** (coloring different parts of the code) make code easier to read.
- **Variables**
  - Variables are like labeled containers used to store information in a program. The value inside a variable can change.
  - **Naming Rules:** Variable names must be unique, should be descriptive, cannot contain spaces, and must begin with a letter.
- **Data Types**
  - Data types are categories of information. You must know the main types:
    - **Integers:** Whole numbers (e.g., 5, -3, 1000).
    - **Floating Point:** Numbers with a decimal point (e.g., 3.14, -2.5).
    - **Strings:** Sequences of text, always in quotes (e.g., "Hello World", "Robot").
    - **Boolean:** Can only be one of two values: `true` or `false`.
- **Input and Output (I/O)**
  - **Input** is any information coming *into* your program, such as a keyboard entry or a sensor reading.
  - **Output** is any result going *out* of your program, like text on a screen, a flashing light, or a motor turning.
- **Making Decisions (If-Then-Else)**

    - Conditional statements use `if-then-else` logic to allow a program to make decisions.
    - They use **comparison operators** (like `==` for equal to, `>` for greater than) and **logical operators** (like `AND` and `OR`) to test conditions.
- **Repetition (Loops)**
    - Loops allow a program to repeat a block of code efficiently.
    - **FOR loop:** Repeats a specific number of times.
    - **WHILE loop:** Repeats as long as a certain condition is true.