CSE 1223: Introduction to Computer Programming in Java
Chapter 1 – Computer Basics
Computer Basics

- Computer system: hardware + software
- Hardware: the physical components
- Software: the instructions that tell the hardware what to do
Common Hardware Components

- **Processor (CPU)**
  - Central Processing Unit
  - Interprets and executes the instructions

- **Memory**
  - main & auxiliary
  - holds data and instructions

- **Input device(s)**
  - mouse, keyboard, etc.

- **Output device(s)**
  - video display, printer, etc.

- CPU and memory are physically housed together

---

![Standard Hardware Organization Diagram](image-url)
Running a Program

*Program*—a set of instructions for a computer to follow
Programming Languages

- Needed to write software
- High-level languages (e.g., Java)
  - relatively easy for people to write and understand
  - not directly understood by computers
- Low-level languages (machine language)
  - directly understood by computer
  - computer-dependent
The Compiler

- A program written in a high-level language (known as the **source code**) cannot be executed directly by the computer.
- A **compiler** is a program that translates source code into machine code that does the same thing (known as the **object code**).
Java Program Translation and Execution

Java Source Code → Java Compiler → Java Byte-Code → Java Virtual Machine

Input to Java Program

Output of Java Program
Java byte-code is portable (hardware-independent)

The Java Virtual Machine (JVM) executes Java byte-code on a real machine

The java compiler is called `javac`

The JVM emulator is called `java`
Algorithmic Thinking

- **Algorithm** - a set of instructions (steps) for solving a problem.
  - must be precise
  - must be complete
  - can be written in an arbitrary notation (e.g., natural language, programming language, diagram, mix, etc.)

- **Algorithmic thinking** is fundamental to computer science and programming
Example of an Algorithm

Algorithm that determines the total cost of a list of items:

1. Write the number 0 on the blackboard.
2. Do the following for each item on the list:
   a. Add the cost of the item to the number on the blackboard.
   b. Replace the old number on the board by this sum.
3. Announce that the answer is the number written on the board.
public class FirstProgram {

    public static void main(String[] args) {

        System.out.println("Hello out there.");
        System.out.println("How’s it going?");
        System.out.println("Good-bye.");
    }

}
Language Syntax

- **Syntax** of a language is a set of (grammar) rules that describe the correct way to write sentences (programs) in the language.

- Programming languages have a very precise syntax: If you break the rules, you’ll get one (or more) errors.
Structure of a Java Program

// import needed libraries

public class ProgramName {

    public static void main(String[] args) {
    
        // statements go here to describe
        // actions to be taken by the program
    
    }

}
A Java Statement

System.out.println("some message here");

- Outputs the message in quotes to the screen (without the quotes)
What Does FirstProgram Do?

- Take a look at the program and see if you can figure out what the program does.
- How do we do this?
- We “trace” the program
  - In other words – go line by line through the program and ask: “What does this line do?”
# First Java Program - Trace

<table>
<thead>
<tr>
<th>Program Line</th>
<th>What we see</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.out.println(&quot;Hello out there.&quot;);</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hello out there.</td>
</tr>
<tr>
<td>System.out.println(&quot;How’s it going?&quot;);</td>
<td>Hello out there.</td>
</tr>
<tr>
<td></td>
<td>How's it going?</td>
</tr>
<tr>
<td>System.out.println(&quot;Good-bye.&quot;);</td>
<td>Hello out there.</td>
</tr>
<tr>
<td></td>
<td>How's it going?</td>
</tr>
<tr>
<td></td>
<td>Goodbye.</td>
</tr>
</tbody>
</table>
What Does FirstProgram Do?

- So based on our trace, what does it do?
- It outputs the following:

  Hello out there.
  How’s it going?
  Good-bye.
import java.util.Scanner;

public class EggBasket
{
    public static void main(String[] args)
    {
        int numberOfBaskets=2;
        int eggsPerBasket=12;
        int totalEggs = numberOfBaskets * eggsPerBasket;

        System.out.println("Total number of eggs is "+ totalEggs);
    }
}
What Does EggBasket Do?

- Take a look at the program and see if you can figure out what it does.
  - Note that unlike our first program, not everything that EggBasket does is done by displaying things to the screen.
  - It does some math as well as displaying things.
Tracing EggBasket

- To trace EggBasket we need to think about what is going on in the computer
  - We call the internal configuration of the program the *program state* – more on this later
  - For now, let’s see how we’ll trace the program
## EggBasket - Trace

<table>
<thead>
<tr>
<th>Program Line</th>
<th>Program state</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int numberOfBaskets=2;</code></td>
<td>numberOfBaskets = 2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><code>int eggsPerBasket=12;</code></td>
<td>numberOfBaskets = 2</td>
</tr>
<tr>
<td></td>
<td>eggsPerBasket = 12</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><code>int totalEggs = numberOfBaskets*eggsPerBasket;</code></td>
<td>numberOfBaskets = 2</td>
</tr>
<tr>
<td></td>
<td>eggsPerBasket = 12</td>
</tr>
<tr>
<td></td>
<td>totalEggs = 24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><code>System.out.println(&quot;Total eggs: &quot;+totalEggs);</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>numberOfBaskets = 2</td>
</tr>
<tr>
<td></td>
<td>eggsPerBasket = 12</td>
</tr>
<tr>
<td></td>
<td>totalEggs = 24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Outputs to screen: [Total eggs: 24]</td>
</tr>
</tbody>
</table>
Programming Errors

- **Syntax** errors—violation of language’s syntax rules, e.g., misspelling a word, forgetting a ;, etc. Caught by the compiler!

- **Runtime** errors—execution errors, e.g., division by zero.

- **Logical** errors—the program compiles and runs without runtime errors, but it does not do what it is supposed to.
import java.util.Scanner;

public class EggBasket {
    public static void main(String[] args) {
        int numberOfBaskets = 2;
        int eggsPerBasket = 12;
        int totalEggs = numberOfBaskets * eggsPerBasket;
        System.out.println("Total number of eggs is "+ totalEggs);
    }
}
import java.util.Scanner;

public class EggBasket {
    public static void main(String[] args) {
        int numberOfBaskets = 2;
        int eggsPerBasket = 0;
        int totalEggs = numberOfBaskets / eggsPerBasket;

        System.out.println("Total number of eggs is "+ totalEggs);
    }
}

Runtime Error! Attempt to divide by zero
Programming Errors

```java
import java.util.Scanner;

public class EggBasket {
    public static void main(String[] args) {
        int numberOfBaskets = 2;
        int eggsPerBasket = 12;
        int totalEggs = numberOfBaskets + eggsPerBasket;
        System.out.println("Total number of eggs is "+ totalEggs);
    }
}
```

Logical Error! To find total number of eggs in all baskets, we need to multiply # of baskets by the eggs per basket, not add the two together!
import java.util.Scanner;

public class EggBasketEnhanced
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);

        System.out.print("Enter the number of eggs in each basket: ");
        String eggsStr = keyboard.nextLine();
        int eggsPerBasket = Integer.parseInt(eggsStr);
        System.out.print("Enter the number of baskets: ");
        String basketStr = keyboard.nextLine();
        int numberOfBaskets = Integer.parseInt(basketStr);

        int totalEggs = numberOfBaskets * eggsPerBasket;

        System.out.println(eggsPerBasket + " eggs per basket.");
        System.out.println(numberOfBaskets + " baskets.");
        System.out.println("Total number of eggs is " + totalEggs);
    }
}
What Does EggBasketEnhanced Do?

- Take a look at the program and see if you can figure out what it does.
  - There are some things in here we haven’t talked about yet.
  - Can you guess what they’re doing based on their names and how we’re using them?
    - Can you trace it?