JAVA PROGRAMMING

Chapter 1

Fundamentals of Java Programming

Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- Elements of a Java Program
- Features of Java
- Accessing the Classes and Class Members
- The Memory Usage by a Java Program
- When Will the Error Occur?

Learning Objectives

- Understand how computers and computer programs work.
- Understand how a Java program is written, compiled, and executed.
- Understand what makes Java platform independent.
- Identify the object-oriented features of Java.
- Identify different elements of a Java program: primitive variable, reference variable, local variable, instance variable, method, and class.

Learning Objectives (cont.)

- Identify where in memory the method invocations, objects, and variables are stored.
- Understand how access modifiers define the accessibility of classes and class members.
- Understand the concepts of early binding and late binding in the context of program errors.

Computers and Computer Programming

- How a Computer Works
- How a Computer Program Works

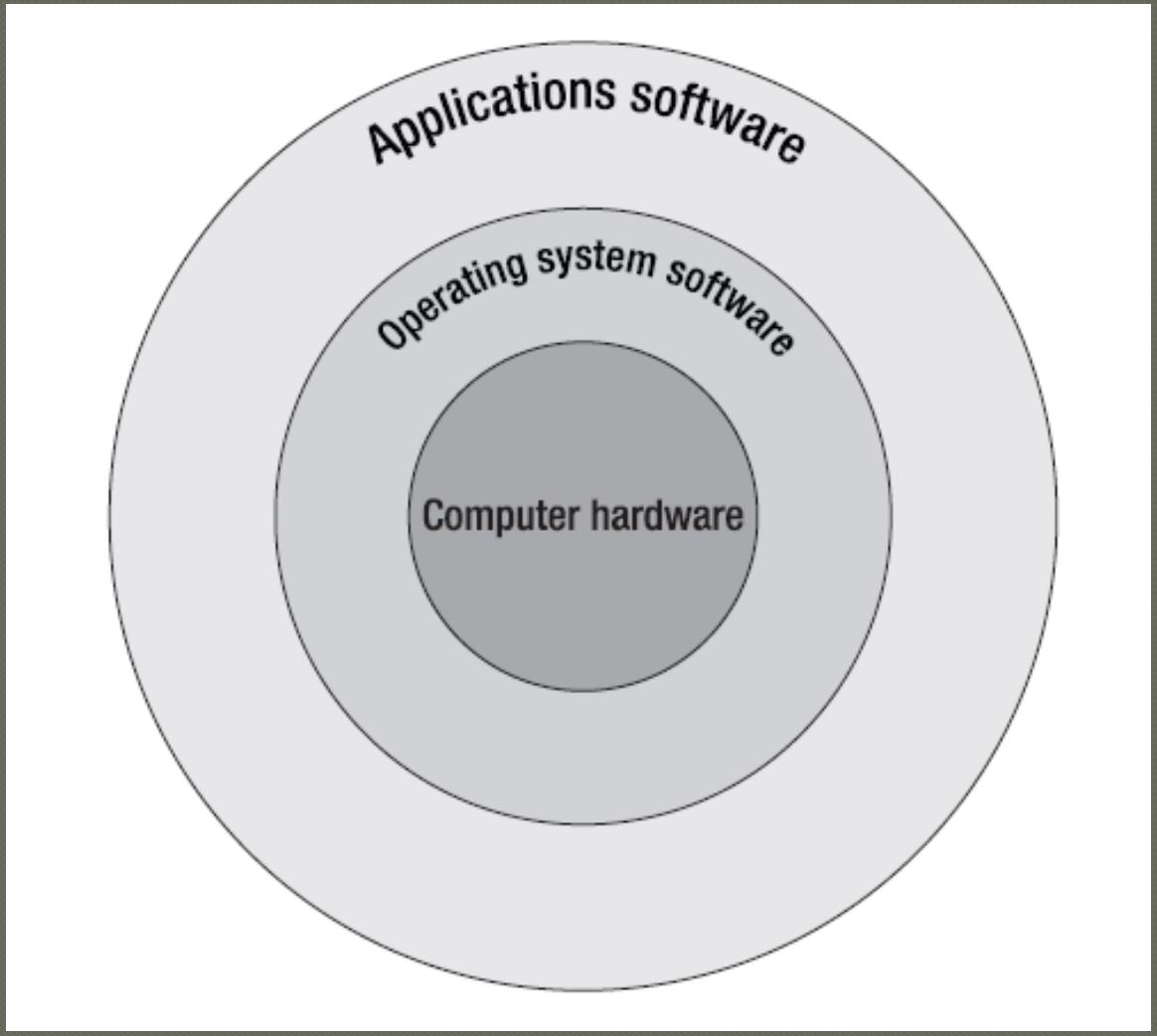
How a Computer Works

- From a computer program's perspective, a computer consists of components to do the following:
  - Receive data from a user
  - Process the data according to instructions from a program or a user
  - Place the results somewhere

How a Computer Works (cont.)

- Places to Store Data:
  - Permanent storage: hard drive
  - Temporary storage: RAM
- I/O Devices: monitor, keyboard, disk, and printer
- CPU: The Brain of the Computer

How a Computer Program Works



Fundamentals of Java Programming

- Computers and Computer Programming
- **Writing and Executing a Java Program**
- Elements of a Java Program
- Features of Java
- Accessing the Classes and Class Members
- The Memory Usage by a Java Program
- When Will the Error Occur?

Writing and Executing a Java Program

- A ***compiler*** translates a program in C/C++ to a binary format called ***executable code***
- The executable code is ***machine dependent***
- Java programs are ***machine independent***, thanks to ***Java virtual machine*** *(JVM).*

Writing and Executing a Java Program

- Writing a Java Program
- Compiling a Java Program
- Executing a Java Program

Writing a Java Program

- ***Source code**:* a set of instructions in text format written according to the rules of the Java programming language.

- ***Source file**:* contains these instructions, and has the file extension .java

Compiling a Java Program

- ***Machine language:*** a binary format
- In Java, the compiler compiles the source code into ***bytecode***
- To create the bytecode files (.class) from the source file RobotManager.java:

```
javac RobotManager.java
```

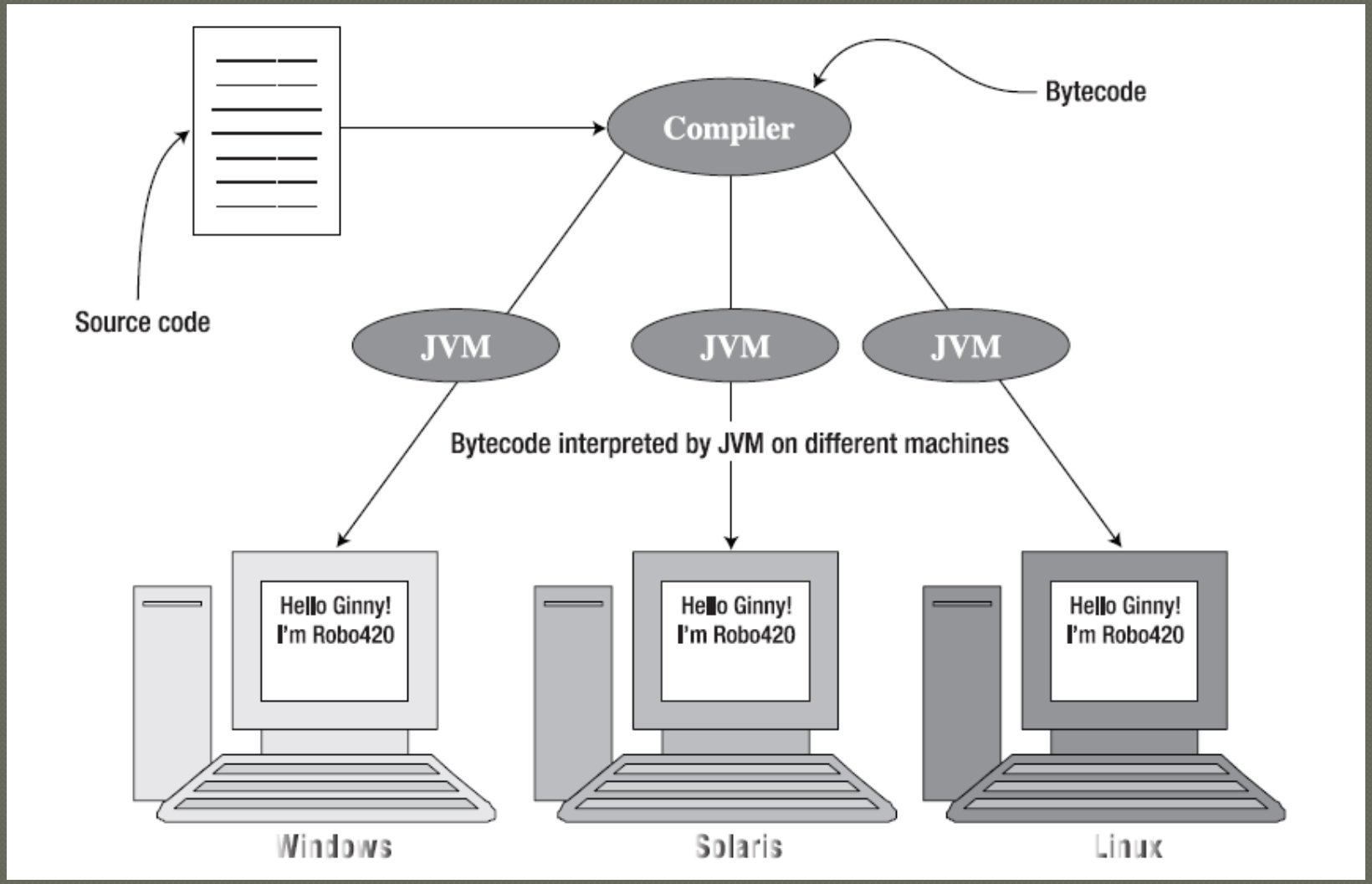
Executing a Java Program

- Executing a Java program by issuing the ***java*** command:

```
java RobotManager Ginny 420
```

- The JVM reads the bytecode file and translates the instructions to the executable format that your computer can understand

write once, run anywhere



Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- **Elements of a Java Program**
- Features of Java
- Accessing the Classes and Class Members
- The Memory Usage by a Java Program
- When Will the Error Occur?

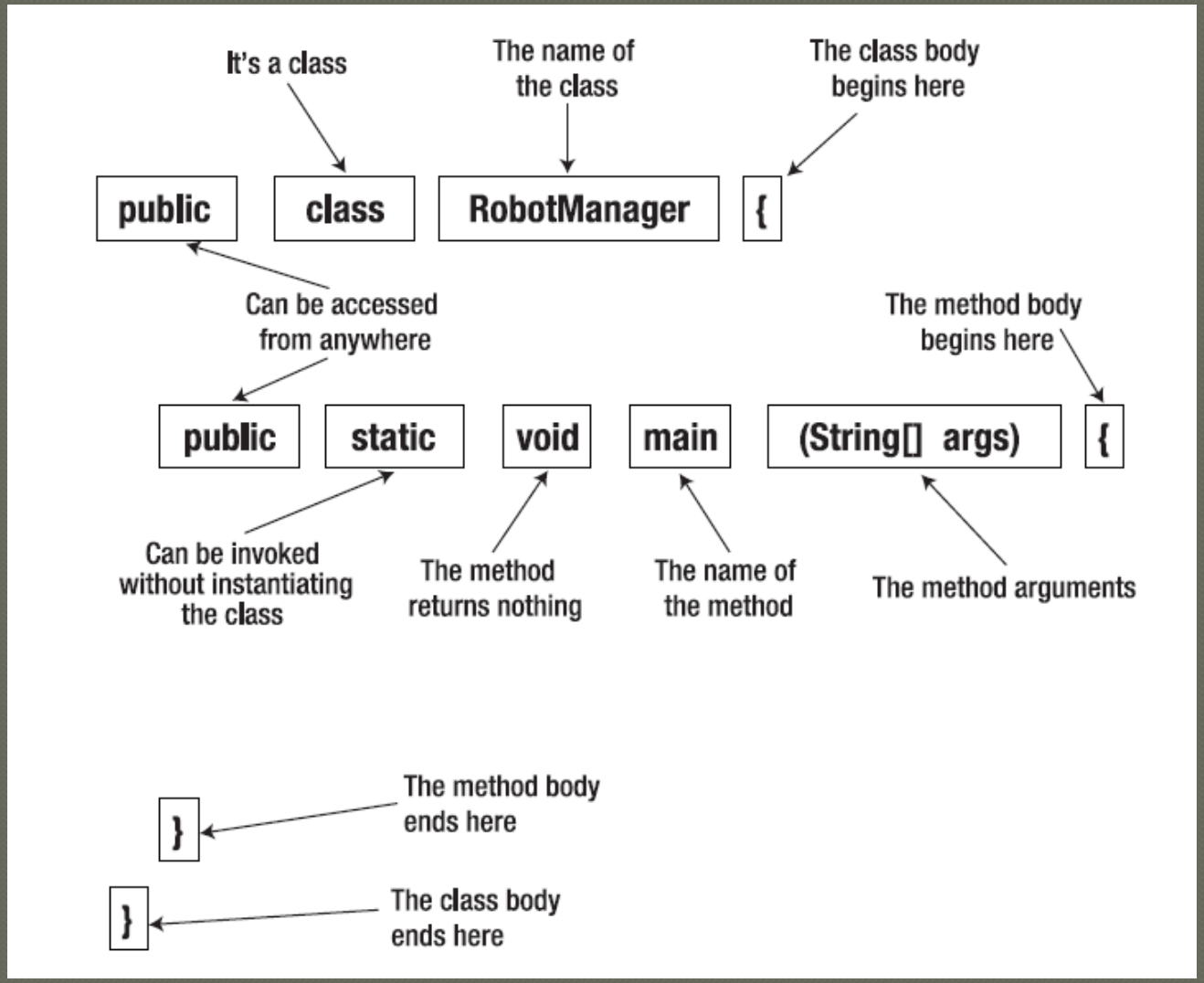
Elements of a Java Program

- Classes and Objects
- Methods
- Variables and Data Types
- Execution Flow of a Program

Classes and Objects

- A class is a template (or a blueprint) from which objects are created
- Writing a class is called ***implementing*** a class: declaration & body
- An object is ***instantiated*** from a class, and also called the ***instance*** of that class
- Each object has a ***state***, a set of characteristics, and a ***behavior*** represented by *methods*

Methods


It's a class
The name of the class
The class body begins here
public
class
RobotManager
{
Can be accessed from anywhere
The method body begins here
public
static
void
main
(String[] args)
{
Can be invoked without instantiating the class
The method returns nothing
The name of the method
The method arguments
}
The method body ends here
}
The class body ends here

Variables and Data Types

- The state of an object is represented by a set of data items that are handled by using variables
- The variable's type determines what kind of values it can hold
- The declaration of a variable:

```
<type> <name>;
```

- An ***object reference variable*** refer to an object:

```
Robot robot;
```

Execution Flow of a Program

- **Expressions**: combination of variables, operators, literals, and method calls
- **Statements**: a complete execution unit of a program; contain one or more expressions
- **Blocks**: a group of zero or more statements between an opening brace and a closing brace
- **Execution Flow Control**: skip, execute, or repeatedly execute a block of statements

Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- Elements of a Java Program
- **Features of Java**
- Accessing the Classes and Class Members
- The Memory Usage by a Java Program
- When Will the Error Occur?

Features of Java

- Platform Independence
- Object-Oriented Programming

Platform Independence

- “write once, run anywhere.”
- The Java compiler compiles the source code into *bytecode*, which can be interpreted by a suitable JVM on any platform
- The JVM can prevent the code from generating side effects outside the system.

Object-Oriented Programming

- **Encapsulation**: combining an object's data with its methods
- **Inheritance**:
  - *Code reusability*
  - *Code maintenance*
  - *Implementing OOP*
- **Polymorphism**: allows an object of a superclass to refer to an object of any subclass.

Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- Elements of a Java Program
- Features of Java
- **Accessing the Classes and Class Members**
- The Memory Usage by a Java Program
- When Will the Error Occur?

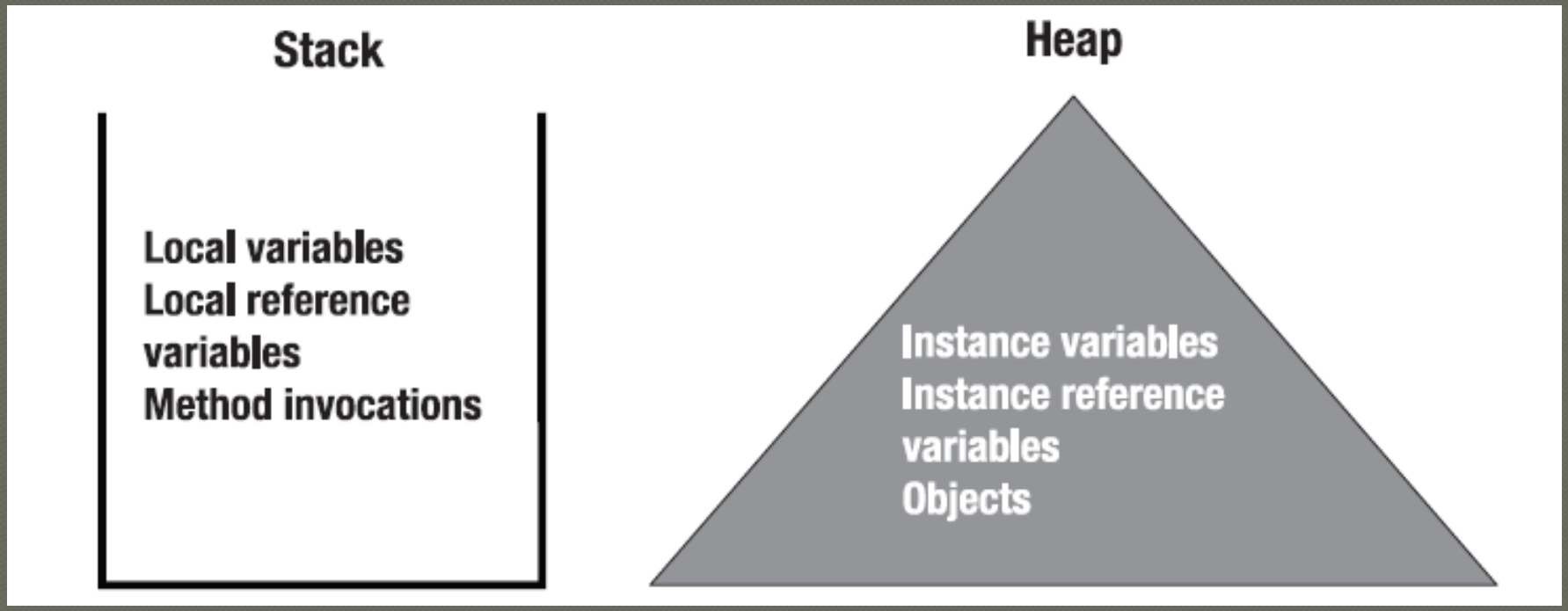
Accessing the Classes and Class Members

- ***Class members***: variables, methods, and nested classes.
- ***Access modifiers***: control the access to the class members
  - **`public`**
  - **`protected`**
  - *default*
  - **`private`**

Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- Elements of a Java Program
- Features of Java
- Accessing the Classes and Class Members
- **The Memory Usage by a Java Program**
- When Will the Error Occur?

The Memory Usage by a Java Program



Fundamentals of Java Programming

- Computers and Computer Programming
- Writing and Executing a Java Program
- Elements of a Java Program
- Features of Java
- Accessing the Classes and Class Members
- The Memory Usage by a Java Program
- **When Will the Error Occur?**

When Will the Error Occur?

- Compilation fails
- An exception is thrown at runtime