A SIMPLE JAVA PROGRAM

Ch 1.2

A Sip of Java: Outline

- History of the Java Language (FYI)
- Applications and Applets (FYI)
- A First Java Application Program
- Writing, Compiling, and Running a Java Program

History of Java

- In 1991, James Gosling and Sun Microsystems began designing a language for home appliances (toasters, TVs, etc.).
  - Challenging, because home appliances are controlled by many different chips (processors)
  - Programs were translated first into an intermediate language common to all appliance processors.
  - Then the intermediate language was translated into the machine language for a particular appliance's processor.
  - Appliance manufacturers weren't impressed.
- In 1994, Gosling realized that his language would be ideal for a Web browser that could run programs over the Internet.
  - Sun produced the browser known today as HotJava.

Applications and Applets

- Two kinds of java programs: applications and applets
- Applications
  - Regular programs
  - Meant to be run on your computer
- Applets
  - Little applications
  - Meant to be sent to another location on the internet and run there
  - Deprecated by Oracle and not supported by many web browsers today in favor of HTML5 and JavaScript

Some Terminology

- The person who writes a program is called the programmer.
- The person who interacts with the program is called the user.
- A package is a library of classes that have been defined already.
  - `import java.util.Scanner;`

Some Terminology

- The item(s) inside parentheses are called argument(s) and provide the information needed by methods.
- A variable is something that can store data.
- An instruction to the computer is called a statement; it ends with a semicolon.
- The grammar rules for a programming language are called the syntax of the language.

A First Java Application

```
import java.util.Scanner;
public class FirstProgram
{   public static void main (String [] args)
    {   System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```

```
Hello out there.
I will add two numbers for you.
Enter two whole numbers on a line:
12 30
The sum of those two numbers is
42
```

Sample screen output

Printing to the Screen

```
System.out.println ("Whatever you want to print");
```

- `System.out` is an object for sending output to the screen.
- `println` is a method to print whatever is in parentheses to the screen.

- The object performs an action when you invoke or call one of its methods

```
objectName.methodName(argumentsTheMethodNeeds);
```

A First Java Application

```
import java.util.Scanner;
public class FirstProgram
{   public static void main (String [] args)
    {   System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```

```
Hello out there.
I will add two numbers for you.
Enter two whole numbers on a line:
12 30
The sum of those two numbers is
42
```

Sample screen output

Compiling a Java Program or Class

- A Java program consists of one or more classes, which must be compiled before running the program.
- You need not compile classes that accompany Java (e.g. `System` and `Scanner`).
- Each class should be in a separate file and ending with `.java`
- The name of the file should be the same as the name of the class.

What would the file for the sample program be named?

Compiling and Running

- Use an *IDE* (integrated development environment) which combines a text editor with commands for compiling and running Java programs.
- When a Java program is compiled, the byte-code version of the program has the same name, but the ending is changed from `.java` to `.class`.

Compiling and Running

- A Java program can involve any number of classes.
- The class to run will contain the words

```
public static void main(String[] args)
```

somewhere in the file

Programming Basics: Outline

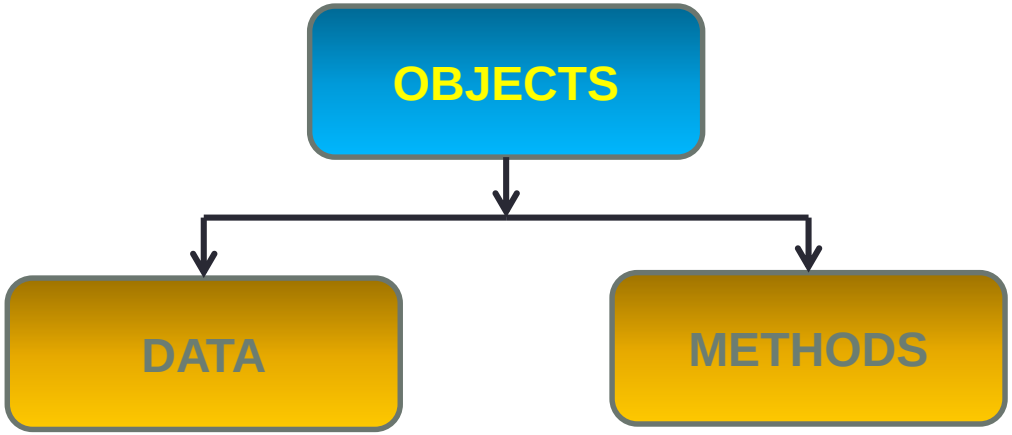
- Object-Oriented Programming

  (briefly only – will be covered later)
- Algorithms
- Testing and Debugging
- Software Reuse

1. PROGRAMMING APPROACHES

Two programming approaches are known:

- The Structured Programming
  - Also known as modular programming.
  - The problem is divided into smaller sub-problems (modules).
  - Each sub-problem is then analyzed and solved.
  - The solutions of all sub-problems are then combined to solve the overall problem.
  - Examples of such programming model languages include Pascal, Fortran and C.
- The Object-Oriented Programming
  - Identify the components of the problem. These are called objects.
  - For each object, identify the relevant data & operations (methods) to be performed on that data.
  - Define the relationship between each object and the other.
  - Examples of programming languages that follow such model are C++ and Java.



Programming

- Programming is a creative process.
- Programming can be learned by discovering the techniques used by experienced programmers.
- These techniques are applicable to almost every programming language, including Java.

Object-Oriented Programming

- Our world consists of objects (people, trees, cars, cities, airline reservations, etc.).
- Objects can perform actions which affect themselves and other objects in the world.
- Object-oriented programming (OOP) treats a program as a collection of objects that interact by means of actions.

OOP Terminology

- Objects, appropriately, are called objects.
- Actions are called methods.
- Objects of the same kind have the same type and belong to the same class.
  - Objects within a class have a common set of methods and the same kinds of data
  - but each object can have it’s own data values.

OOP Design Principles

- OOP adheres to three primary design principles:
  - Encapsulation
  - Polymorphism
  - Inheritance

Algorithms

- By designing methods, programmers provide actions for objects to perform.
- An algorithm describes a means of performing an action.
- Once an algorithm is defined, expressing it in Java (or in another programming language) usually is easy.

Algorithms

- An algorithm is a set of instructions for solving a problem.
- An algorithm must be expressed completely and precisely.
- Algorithms usually are expressed in English or in pseudocode.

Example: Total Cost of All Items

- Write the number 0 on the whiteboard.
- For each item on the list
  - Add the cost of the item to the number on the whiteboard
  - Replace the number on the whiteboard with the result of this addition.
- Announce that the answer is the number written on the whiteboard.

Testing and Debugging

- Eliminate errors by avoiding them in the first place.
  - Carefully design classes, algorithms and methods.
  - Carefully code everything into Java.
- Test your program with appropriate test cases (some where the answer is known), discover and fix any errors, then retest.

Errors

- An error in a program is called a bug.
- Eliminating errors is called debugging.
- Three kinds or errors
  - Syntax errors
  - Runtime errors
  - Logic errors

Syntax Errors

- Grammatical mistakes in a program
  - The grammatical rules for writing a program are very strict
- The compiler catches syntax errors and prints an error message.
- Example: using a period where a program expects a comma

Runtime Errors

- Errors that are detected when your program is running, but not during compilation
- When the computer detects an error, it terminates the program and prints an error message.
- Example: attempting to divide by 0

Logic Errors

- Errors that are not detected during compilation or while running, but which cause the program to produce incorrect results
- Example: an attempt to calculate a Fahrenheit temperature from a Celsius temperature by multiplying by 9/5 and adding 23 instead of 32

Reusable Components

- Most programs are created by combining components that exist already. They are not usually created entirely from scratch.

- Reusing components saves time and money.
- Reused components are likely to be better developed, and more reliable.
- New components should designed to be reusable by other applications.

Software Reuse

- Reusable classes are used
  - Design class objects which are general
  - Java provides many classes
  - Note documentation on following slide

  - https://docs.oracle.com/javase/8/docs/index.html
  - https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html

Software Reuse

FIGURE 1.5 The Documentation for the Class Scanner