

SFWR ENG/COMP SCI 2S03

Principles of Programming

Dr. Ridha Khedri

Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

Acknowledgments: Material based on *Java actually: A Comprehensive Primer in Programming* (Chapter 1)

Outline of Today's Lecture

(Slide 2 of 42)

- 1 Introduction
- 2 Programming
- 3 Editing source code
- 4 Development tools for Java
- 5 Compiling Source Code
- 6 Running Java programs
- 7 The Java Virtual Machine
- 8 The components of a program
 - Operations
 - Principle of Low Coupling and High Cohesion
 - Programming with objects
- 9 The Java programming language
 - Classes and methods
 - Program entry point
 - Variables

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Outline
Today's Lecture

Part I

Review of Previous Lecture

Part II

Today's Lecture

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Introduction

(Slide 5 of 42)

In this lecture we answer the following:

- How to create and maintain a program?
- What source code is and how we create it?
- What the basic components of a Java program are?
- How to compile Java source code into an executable program?
- How to run a compiled Java program?

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Programming

(Slide 6 of 42)

- A program is a planned sequence and combination of instructions designed to achieve specified goals and that can be executed on a computer
- The task of conceiving and writing a new program is called programming
- We write a program using the grammar of the programming language
- The obtained written text is called source code and it is stored in a **text file**
- **The source code might contain as well human-readable descriptions of the task to be performed by the computer**

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

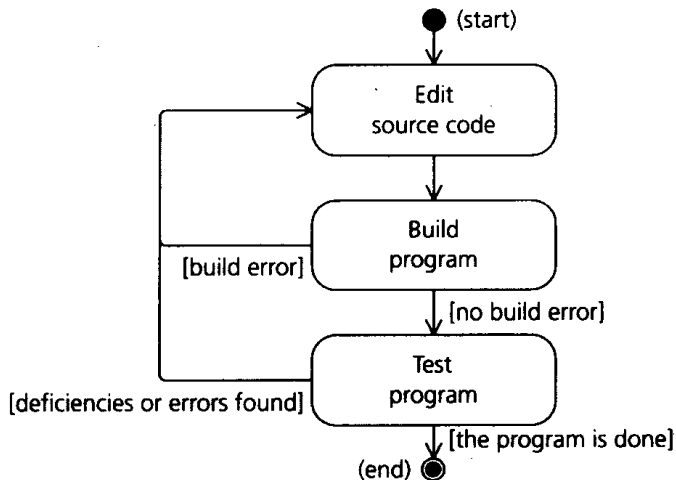
The Java
programming

Programming Overview

Programming

(Slide 7 of 42)

Main activities in writing programs



SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Programming

(Slide 8 of 42)

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

```
1 // (1) This source code file is called SimpleProgram.java
  public class SimpleProgram {
3   // Print a proverb, and the number of characters in the proverb.
   public static void main(String[] args) { // (2)
5       System.out.println("A proverb:"); // (3)
7       String proverb = "Practice makes perfect!"; // (4)
9       System.out.println(proverb); // (5)
11      int characterCount = proverb.length(); // (6)
       System.out.println("The proverb has " + characterCount + "
           characters.");
13  }
  }
```

Listing 1: The source code for a simple Java program

Programming Overview

Programming

(Slide 9 of 42)

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

```
//(1) This source code file is called SimpleProgram.java
2 public class SimpleProgram {
  // Print a proverb, and the number of characters in the proverb.
4  public static void main (String [] args) { // (2)
    System.out.println ("A proverb:"); // (3)
    String proverb = " Practice makes perfect!"; // (4)
    System.out.println (proverb); // (5)
    int characterCount = proverb.length(); // (6)
    System.out.println ("The proverb has " + characterCount +
      " characters.");
  }
14 }
```

Listing 2: The source code for a simple Java program (blanks highlighted)

Programming Overview

Programming

(Slide 10 of 42)

Strong Recommendation: Adhere to “The Elements of Java Style” by Vermeulen et al.

- General principles for program writing
- Formatting conventions
- Naming conventions
- Documentation conventions
- Programming conventions (e.g., type safety, exception handling, concurrency)
- Packaging conventions

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Editing source code

(Slide 11 of 42)

- Source code files contain characters that constitute the actual text of the program (or its elements)
- Word processors are not suitable to write programs (create formatted documents)
- Source code file naming rules
 - The compiler requires the source code files to be named according to specific rules
 - A Java source code file usually contains **a class**
 - **The name of the class is important when naming its corresponding source code file**
 - **ClassName.java (CustomerAccount.java)**
- The compiler refuses to compile files whose names do not have the correct extension

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

- Compiling a program depends on the development tool you are using
- Sun Microsystems Java Development Kit (JDK)
- Java development on any platform often benefits from the use of an Integrated Development Environment (IDE)
 - Eclipse IDE for Java Developers (<http://www.eclipse.org>) –free–
 - Netbeans IDE (<http://www.netbeans.org>) –free–
 - JetBrains IntelliJ IDEA (<http://www.jetbrains.com/idea/>) –requires a license–
 - Xcode (<http://developer.apple.com/tools/xcode/>) –free–
 - MANY MORE

Programming Overview

Compiling Source Code

(Slide 13 of 42)

- To compile a Java program from command line:

```
> javac SimpleProgram.java
```

- The javac compiler reads the source code from the Java source code file
 - Translates each class into a compiled form
 - The obtained file is known as java byte code
 - It creates files named FileName.class
 - It may detect errors in the source code
 - It reports any errors and terminates the compilation

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

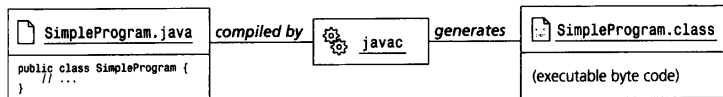
The Java
programming

Programming Overview

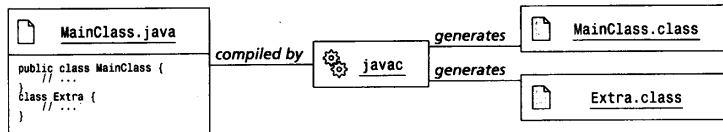
Compiling Source Code

(Slide 14 of 42)

Compiling source code



(a) One class in the source code file



(b) Two classes in the source code file

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Running Java programs

(Slide 15 of 42)

- The command for running `FileName.class` is `java`
- It should not be confused with the command `javac`
- The syntax for running a Java program from the command line is:

```
> java -ea SimpleProgram
```

- `java` starts the Java virtual machine
- `-ea` is a switch that enables assertions
 - you can use `-enableassertions` instead
 - to have more other switches
 - to get them use

```
> man java
```

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Running Java programs

(Slide 16 of 42)

Terminal window showing programm execution

```
C:\programming\java\source>javac SimpleProgram.java
C:\programming\java\source>java -ea SimpleProgram
A proverb:
Practice makes perfect!
The proverb has 23 characters.
C:\programming\java\source>
```

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

Running Java programs

(Slide 17 of 42)

- The java command requires that the exact name of the class containing the main() method is given
- Guidelines for running the java command:
 - Specify the exact class name, without any .class or .java extensions
 - Check the use of upper and lowercase letters in the class name
 - Make sure that already have a .class file for each class in the program

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

**Running Java
programs**

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 18 of 42)

What is a Virtual Machine?

- It is built on an existing system
- It separates a programming language, hardware language, or application from a physical execution platform
- It plays the role of an emulation software
 - It provides an emulation of the functions of one system using a different system
 - It allows exact reproduction of external behavior of a system

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

**The Java Virtual
Machine**

The components of
a program

The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 19 of 42)

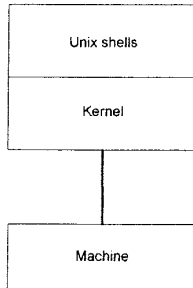


Figure: Unix virtual machine

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

**The Java Virtual
Machine**

The components of
a program

The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 20 of 42)

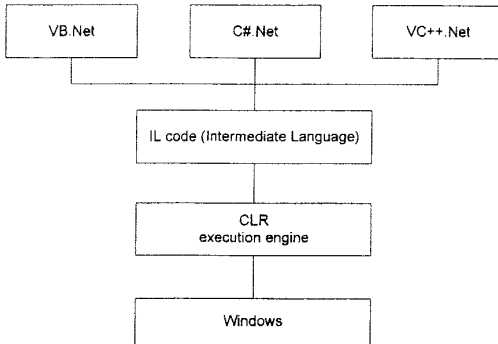


Figure: Common Language Runtime (CLR) virtual machine in .NET platform

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 21 of 42)

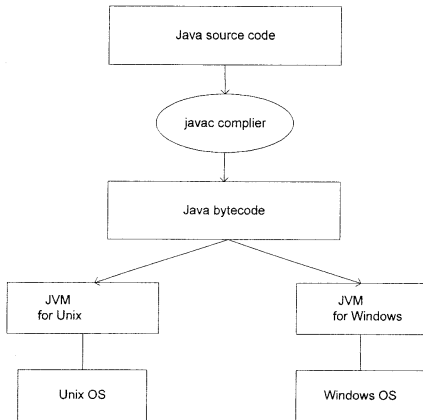


Figure: Java virtual machine

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 22 of 42)

- Applicable Design Domain
 - Solving a problem by simulation or translation
 - Interpreters of microprogramming, XML processing, script command language execution, rule-based system execution, Small talk and Java interpreter typed programming language
- Benefits
 - Portability and machine platform independence
 - Simplicity of the software development
 - Simulation for non-native model
- Limitations
 - Slow execution of the interpreter
 - Additional overhead due to the new layer

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

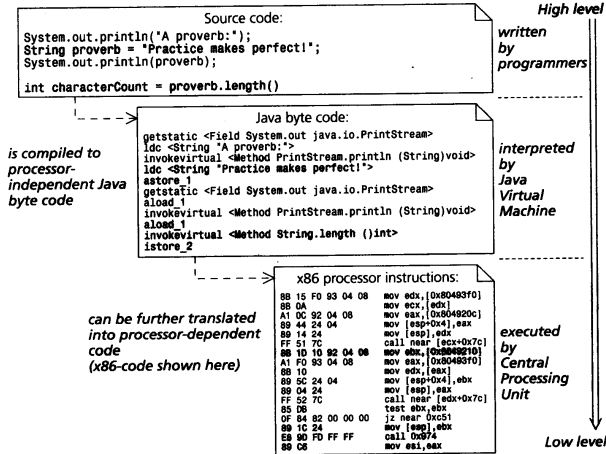
The Java
programming

Programming Overview

The Java Virtual Machine

(Slide 23 of 42)

Program code at several levels



SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction
Programming
Editing source code
Development tools
for Java
Compiling Source
Code
Running Java
programs
The Java Virtual
Machine
The components of
a program
The Java
programming

Programming Overview

The components of a program Operations

(Slide 24 of 42)

- We break down the given problem into smaller tasks
- Each operation is realised by a sequence of actions
- These actions are written in the chosen programming language
- We will be guided by the principle of Low Coupling and High Cohesion

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations

Principle of Low
Coupling and High

Cohesion

In general:

- Cohesion within a module is the degree to which communication takes place among the module's elements
- Coupling describes the degree to which modules depend directly on other modules
- Effective modularization is accomplished by maximizing cohesion and minimizing coupling
- This principle helps to decompose complex tasks into simpler ones

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations
Principle of Low
Coupling and High

Programming Overview

The components of a program

Principle of Low Coupling and High Cohesion

(Slide 26 of 42)

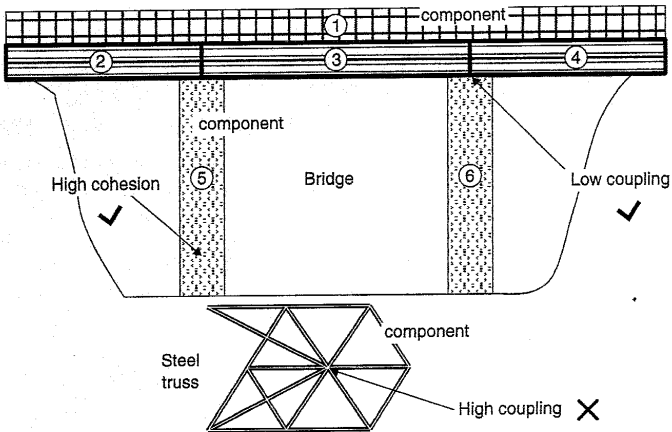


Figure: Cohesion and Coupling

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations
Principle of Low
Coupling and High

In the context of OO Design:

- A system with highly inter-dependable classes is very hard to maintain
- A change in one class may result in cascading updates of other classes
- We should avoid tight-coupling of classes (Identified using analysis class diagram)
- A pair of classes which has dependency association on each other is called tightly-coupled
- Tight coupling might be removed by introducing new classes or inheritance

- To complete a task often we need to involve several types of objects
- Each object has a set of operations
- Example: The operations of making an omelette:
 - ① Open the refrigerator
 - ② Take out an egg carton
 - ③ Open the egg carton
 - ④ Take out two eggs
 - ⑤ Close the egg carton
 - ⑥ Place the egg carton back in the refrigerator
 - ⑦ Close the refrigerator
 - ⑧ Turn on the stove
 - ⑨ ...

Programming Overview

The components of a program

Programming with objects

(Slide 29 of 42)

- Object-based programming (OBP) involves describing tasks in terms of operations
- The operations are executed on objects
- What the objects represent depends on what the program is trying to accomplish
- Programs usually have more than one object of the same type
- Objects that share a set of common properties can be considered to belong to the same class of objects

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

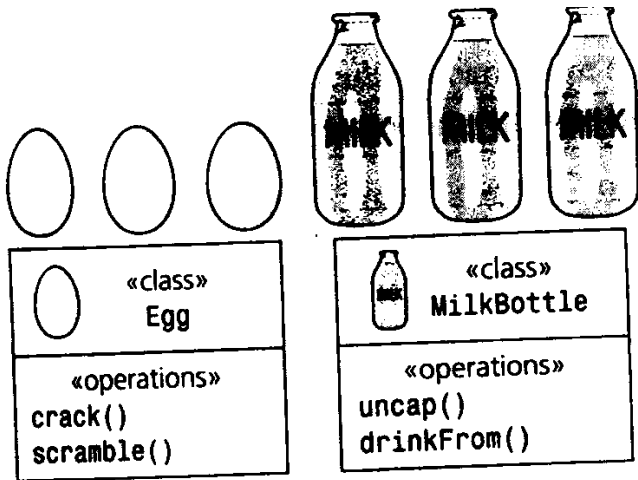
Operations
Principle of Low
Coupling and High

Programming Overview

The components of a program Programming with objects

(Slide 30 of 42)

Classes and instances



SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations
Principle of Low
Coupling and High

Let us ask ourselves some questions

- What could be the main object (then ... class) (the one that should contain the main() method)?
- How can we identify the other classes?
- Can we see the operations of each class?
- Can we see relationships among the identified classes?

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations
Principle of Low
Coupling and High

Programming Overview

The components of a program Programming with objects

(Slide 32 of 42)

- A class is defined by describing
 - the properties specific to the objects of the class (ATTRIBUTES)
 - the operations that can be performed on these objects (METHODS)
- A program can consist of user-defined classes as well as classes from other sources
- The Java language comes with a large collection of ready-to-use classes called the Java standard library
- These classes contain program code that can readily be used

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

Operations
Principle of Low
Coupling and High

SessionController
-state : int = WAIT_LOG_ON
-WAIT_LOG_ON : int = 1
-READY : int = 2
-HANDLING_ADD_ITEM : int = 3
-HANDLING_DEL_ITEM : int = 4
-HANDLING_CLEAR_CART : int = 5
-refInventoryDB : object = null
-refCustomerInfoDB : object = null
-refCustomerInformation : object = null
-shoppingCart : object = null
-refShippingDeptWrapper : object
-refFinancialDeptWrapper : object
+constructor()
+destructor()
+add_item() : string
+del_item() : string
+clear_cart() : string
+log_in() : string
+log_out() : string
+check_out() : string

Figure: Example of the representation of a class

- The language constructs of the Java programming language have a prescribed structure (captured by a GRAMMAR)
- We call this structure the syntax of the language
- A class declaration is a language construct to define classes
- Operations in a class are defined by method declarations
 - It containing sequences of statements describing the actions that need to be performed

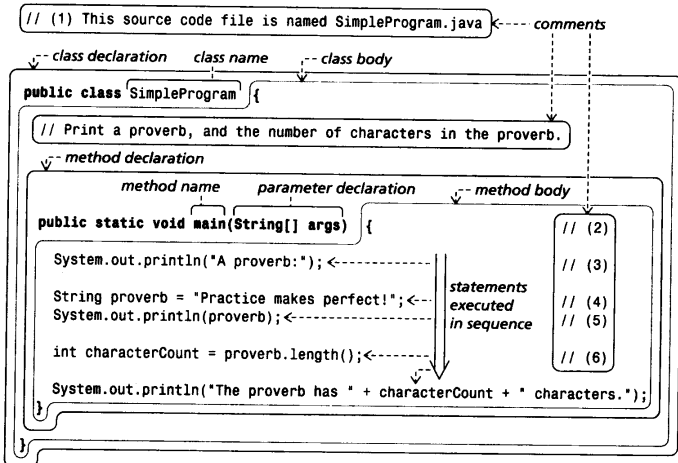
Programming Overview

The Java programming language Classes and methods

(Slide 35 of 42)

Class and method declarations

top of source code file



bottom of source code file

SFWR
ENG/COMP SCI
2S03
Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

For the compiler, the main() method is the following:

```
public static void main(String[]args){System.out.println("A proverb:");  
String proverb="Practice makes perfect!";System.out.println(proverb);int  
characterCount=proverb.length();System.out.println("The proverb has "+  
characterCount+" characters.");}
```

- So, why we add all the rest in our programs?
 - Have readable programs
 - Have maintainable programs
 - Have self-documented programs
 - It is an issue of **sustainability** of our programs (reusable, can last longer)

Programming Overview

The Java programming language

Program entry point

(Slide 37 of 42)

```
public class SimpleProgram {
2 // Line (2) is the PROGRAM ENTRY POINT
  public static void main(String [] args) { // (2)
4 // The part "String [] args" is the parameter declaration

6 System.out.println("A proverb:"); // (3)
    The above statement is the first to be executed

8 String proverb = "Practice makes perfect!"; // (4)
    The above statement is the next to be executed
9 System.out.println(proverb); // (5)

10 int characterCount = proverb.length(); // (6)
    DO NOT FORGET THAT ; is an OPERATOR

12 System.out.println("The proverb has " + characterCount + "
    characters.");
14 } // End of the method main(String [] args)
}
```

Listing 3: The source code for a simple Java program

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

Programming Overview

The Java programming language

Program entry point

(Slide 38 of 42)

Syntax of statements in previous program

object reference *parameter value*

```
System.out.println("A proverb:");
```

method name *method call*

variable name *string value*

```
String proverb = "Practice makes perfect!";
```

variable declaration *variable assignment*

```
System.out.println(proverb);
```

method call

variable declaration *method call*

```
int characterCount = proverb.length();
```

variable assignment

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

- Variables are named locations in the computer's internal storage (memory)
- In a variable, values can be held during program execution
- Methods often use variables to hold intermediate results
- There are several types of values that can be stored in variables (WE WILL EXPLORE THEM LATER)
- When we assign a value to a variable, we store the value in the variable

Programming Overview

The Java programming language

Variables

(Slide 40 of 42)

SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

```
1 public class SimpleProgram {
2     public static void main(String[] args) { // (2)
3
4         System.out.println("A proverb:"); // (3)
5
6         String proverb = "Practice makes perfect!"; // (4)
7         The above statement declares a variable of type string
8         System.out.println(proverb); // (5)
9         The above statement prints the string stored in the variable
10        proverb
11
12        int characterCount = proverb.length(); // (6)
13        The above statement declares a variable of type integer
14
15        System.out.println("The proverb has " + characterCount + "
16        characters.");
17    }
18 }
```

Listing 4: The source code for a simple Java program

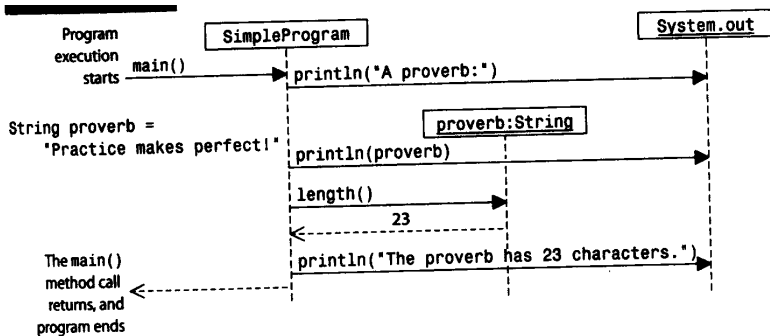
Programming Overview

The Java programming language

Variables

(Slide 41 of 42)

Sequence of method calls during program execution



SFWR
ENG/COMP SCI
2S03

Principles of
Programming

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming

**SFWR
ENG/COMP SCI
2S03**

**Principles of
Programming**

Dr. R. Khedri

Introduction

Programming

Editing source code

Development tools
for Java

Compiling Source
Code

Running Java
programs

The Java Virtual
Machine

The components of
a program

The Java
programming