Building Java Programs

Chapter 1
Lecture 1-1: Introduction; Basic Java Programs

readings: 1.1 - 1.3
self-check: #1-14
exercises: #1-4

What is CSE?

- Computer Science
  - The study of theoretical foundations of information and computation and their implementation and application in computer systems. — Wikipedia
  - Many subfields
    - Graphics, Computer Vision
    - Artificial Intelligence
    - Scientific Computing
    - Robotics
    - Databases, Data Mining
    - Computational Linguistics, Natural Language Processing ...
- Computer Engineering
  - Overlap with CS and EE; emphasizes hardware

The CS job market

What is programming?

- program: A set of instructions to be carried out by a computer.
- program execution: The act of carrying out the instructions contained in a program.
- programming language: A systematic set of rules used to describe computations in a format that is readable by humans.
  - This textbook teaches programming in a language named Java.

Programming languages

- Some influential ones:
  - FORTRAN
    - science/engineering
  - COBOL
    - business data
  - assembler
    - low-level programming
  - BASIC
    - simple language

Some modern languages

- procedural languages: programs are a series of commands
  - Pascal (1970): designed for education
  - C (1972): low-level operating systems and device drivers
  - functional programming: functions map inputs to outputs
  - object-oriented languages: programs use interacting "objects"
    - Smalltalk (1980): first major object-oriented language
    - Java (1995): "object-oriented" improvements to C
      - "object-oriented" improvements to C
        - used in major industry, used to build major class such as Android
  - Java (1995): designed for embedded systems, web apps/servers
    - Runs on many platforms (Windows, Mac, Linux, cell phones ...)
  - The language taught in this textbook
Basic Java programs with println statements

.reading: 1.2 - 1.3
.self-check: #5-14
.exercises: #1-4

A Java program

```java
public class Hello {
    public static void main(String[] args) {
        System.out.println("hello, world");
        System.out.println("this program produces
        first lines of output");
    }
}
```

- its output:
  - hello, world
  - This program produces
  - first lines of output
- console: Text box into which the program's output is printed.

System.out.println

- A statement that prints a line of output on the console.
- pronounced "print-line"
- sometimes called a "println statement" for short
- Two ways to use System.out.println:
  - System.out.println("text");
    Prints the given message as output.
  - System.out.println();
    Prints a blank line of output.

Compiling/running a program

1. Write it.
   - code or source code: The set of instructions in a program.
2. Compile it.
   - compile: Translates a program from one language to another.
   - byte code: The Java compiler converts your code into a
     format named byte code that runs on many computer types.
3. Run (execute) it.
   - output: The messages printed to the user by a program.

Structure of a Java program

```java
public class Name {
    class a program
    public static void main(String[] args) {
        statement.
    }
    statement:
    statement:
    method: a named group of statements
    statement: a command to be executed
```

- Every executable Java program consists of a class,
  - that contains a method named main,
  - that contains the statements (commands) to be executed.

Names and identifiers

- You must give your program a name.
  - public class Greetings {
  - Naming convention: capitalize each word (e.g., Greetings)
  - Your program's file must match exactly (Greetings.java)
  - includes capitalization (Java is "case-sensitive")
- Identifier: A name given to an item in your program.
  - must start with a letter or _
  - subsequent characters can be any of these or a number
  - illegal: name. There is: ADDED_TO_VAR Tailings
  - illegal: start. This is: start_of

```
Keywords

- **keyword**: An identifier that you cannot use because it already has a reserved meaning in Java.

  - abstract
  - default
  - if
  - implements
  - instanceof
  - throw
  - public
  - throws
  - return
  - super
  - this
  - switch
  - try
  - catch
  - final
  - interface
  - static
  - void
  - char
  - float
  - native
  - short
  - var
  - class
  - enum
  - native
  - volatile
  - boolean
  - double
  - new
  - this
  - while

Syntax error (compiler error): A problem in the structure of a program that causes the compiler to fail.

- **missing semicolon**
- **too many or too few **
- **keywords**
- **illegal identifier**
- **class and file names do not match**

Syntax error example

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

**Compiler output:**

- Hello.java:1: syntax error expected
- Hello.java:3: String[] expected
- compile failed
- 2 errors

- The compiler shows the line number where it found the error.
- The error messages can be tough to understand!

Strings

- **string**: A sequence of characters to be printed.
  - Must start and end with a ‘”’ (quote) character.
  - The quotes do not appear in the output.

- **Examples:**
  - "Hello"
  - "This is a string. It's very long!"

- **Restrictions:**
  - May not span multiple lines.
  - "This is not a legal String.

- May not contain a ‘”’ character.
  - "This is not a "String" string either."

Escape sequences

- **escape sequence**: A special sequence of characters used to represent certain special characters in a string.

  - \t: tab character
  - \n: new line character
  - \": quotation mark character
  - \\: backslash character

- **Example:**
  ```java
  System.out.println("\tHello\nWorld\n\"");
  System.out.println("\\\n\n\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n```
**Answers**

1. **Output of each `println` statement:**
   ```java
   a b c
   ///
   ///
   ///
   ///
   ///
   ///
   ///
   ///
   ///
   ///
   ///
   in he downward spiral
   ```

2. **`println` statement to produce the line of output:**
   ```java
   System.out.println("a b c");
   System.out.println("///");
   ```

**Questions**

1. **What `println` statements will generate this output?**
   This program prints a quote from the Gettysburg Address.
   "Four score and seven years ago, our 'forefathers' brought forth on this continent a new nation..."

2. **What `println` statements will generate this output?**
   A "spread" string is printed better if you learn the rules of "escape sequences."
   Also, "" represents an empty string, don't forget! use "" instead of ""!
   "Is not the same as!"