Computational Expression

Computer and Java Basics

Janyl Jumadinova

23 January, 2019
What is Computer Programming?

Programming is the act of writing usable and useful software. A program is a set of instructions.
What is Computer Programming?

- Programming is the act of writing usable and useful software
- A program is a set of instructions
Programming Language

- We will use **Java** programming language in this class.
- Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun’s Java platform.
HISTORY OF JAVA

- Started development in 1991 at Sun
- Originally called Oak
- Intended for smart consumer-electronic devices
- Derives much syntax/concepts from C++
  - BCPL → B → C → C++ → Java
- Development almost halted, but 1993 saw introduction of web; Java was revamped to be able to easily add dynamic content to web pages
- Formally announced and released in May 1995
- Released under GPL to the public in May 2007
Java is an object-oriented programming language.
Programming in Java

- Java is an *object-oriented* programming language
- **Objects** are fundamental elements that make up a program
Programming in Java

- Java is an **object-oriented** programming language
- **Objects** are fundamental elements that make up a program
- Java has a library of software, called **Java API**, that is available for your use
Java program development process

MyProgram.java → Compiler → MyProgram.class → Java VM → 0100101... → My Program
/** This is the first program people write in a new language, the "Hello World!". In Java, this file must be named Welcome.java, with the first part of the name, Welcome, being the same as the name of the class. The filename itself (not the class name) must always end in .java to indicate to the operating system that it’s a java source file. */

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

Comments in Java can be one of three styles:

- **Single line:** starts at // anywhere on a line, ends at the end of that line

- **Multi-line:** starts with character sequence /* anywhere, ends with character sequence */ anywhere after that can span multiple lines

- **javadoc:** starts with character sequence /** anywhere, ends with character sequence */ anywhere, after that uses javadoc utility to create HTML documentation from code
public class Welcome:

- **public** means that something is available across packages (reserved word)
- Name of the class has to be the same as the name of the .java file
public class Welcome:

- **public** means that something is available across packages (reserved word)
- Name of the class has to be the same as the name of the .java file

**public static void main ( String identifier[] ):**

- The particular form of main is required by Java.
- JVM starts executing here!
- main is a static method, it is part of its class and not part of objects.
- Strings in Java are sequence of characters
public class Welcome:

- **public** means that something is available across packages (reserved word)
- Name of the class has to be the same as the name of the .java file

public static void main ( String identifier[] ):

- The particular form of main is required by Java.
- JVM starts executing here!
- main is a static method, it is part of its class and not part of objects.
- Strings in Java are sequence of characters

- Braces \{ \} are used to collect statements into a "block"
public class Welcome:

- public means that something is available across packages (reserved word)
- Name of the class has to be the same as the name of the .java file

public static void main ( String identifier[]) :

- The particular form of main is required by Java.
- JVM starts executing here!
- main is a static method, it is part of its class and not part of objects.
- Strings in Java are sequence of characters

- Braces { } are used to collect statements into a "block"
- Statements in Java end with semicolons.
Printing

- `println`: New line after printing
- `print`: No new line
- `printf`: Can specify format - may learn this later
string literal in class `String`
“ABC”
“This is interesting”
“ ”
“91”
Character Strings

string literal in class `String`
“ABC”
“This is interesting”
“ ”
“91”

- Use `print` or `println` methods to print a character string to the terminal
- `System.out.println(“CMPSC 100”);`
- the string “CMPSC 100” is a parameter: data sent to a method
String Concatenation

Appending one string to the end of another: use + operator
“This is ” + “interesting”
“Your grade is ” + “91”
String Concatenation

Appending one string to the end of another: use + operator
“This is ” + “interesting”
“Your grade is ” + “91”

+ is also used for arithmetic addition

System.out.println(”Adding ” + 12 + 23); is not the same as
System.out.println(”Adding ” + (12 + 23));
Escape sequences, or escape characters, begin with a slash and are immediately followed by another character.

This two-character sequence, inside " " allows you to control your output (\n, \t, \b) or output characters you wouldn’t otherwise be able to (\\, \") inside a string.
## Escape Sequences

<table>
<thead>
<tr>
<th>Seq</th>
<th>Meaning</th>
<th>Example Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>\n</td>
<td>New line</td>
<td>System.out.println(”Hi\nThere”);</td>
</tr>
<tr>
<td>\t</td>
<td>Horizontal tab</td>
<td>System.out.println(”What’s\tup?”);</td>
</tr>
<tr>
<td>\b</td>
<td>Backspace</td>
<td>System.out.println(”Hi\b Hey”);</td>
</tr>
<tr>
<td>\</td>
<td>Backslash</td>
<td>System.out.println(”Back\Slash”);</td>
</tr>
<tr>
<td>\”</td>
<td>Double quote</td>
<td>System.out.println(”Dbl\Quote”);</td>
</tr>
</tbody>
</table>