Basics of Java Programming
- Strings and Printing

CSC 1051 – Algorithms and Data Structures I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus
Lab 1:

- Learn about jGrasp - the programming environment that we will be using in this class
  - Compile and run a java program
- Understand the relationship between a Java class name and the name of the .java file where the class is defined
- Practice using basic Java output statements and adding comments
- Learn about variables, string literals, concatenation. *E.g.*,
  ```java
  System.out.println ("Howdy " + name);
  System.out.println ("The answer is " + x);
  System.out.print ("Counting... up: " + (count + 1));
  System.out.println (" ... and\n ... down: " + (count - 1));
  ```
- Explore Java syntax
- Experience some errors!
Character Strings

• A string literal is represented by putting double quotes around the text

• Examples:

  "This is a string literal."
  "123 Main Street"
  "X"
Character Strings

- A *string literal* is represented by putting double quotes around the text

- Examples:

  "This is a string literal."
  "123 Main Street"
  "X"

spaces matter in here!
The println Method

• In the Lincoln program we invoked the println method to print a character string

• The System.out object represents a destination (the monitor screen) to which we can send output

```java
System.out.println("Whatever you are, be a good one.");
```

object  method  name

information provided to the method (parameters)
The print Method

• In the *Lincoln* program we invoked the `println` method to print a character string.

• The `System.out` object represents a destination (the monitor screen) to which we can send output.

• `print` is similar to the `println` except that it does not advance to the next line.

```
System.out.print ("Whatever you are, be a good one.");
```
String Concatenation

- The *string concatenation operator* (+) is used to append one string to the end of another

  "And one more " + "thing"
Hands on:

• Use MyQuote.java as a starting point (program from Lab 1), focus on this part of the code:

```
System.out.println("Howdy " + name);
System.out.println("The answer is " + x);
System.out.print("Counting... up: " + (count + 1));
System.out.println(" ... and\n ... down: " + (count - 1));
```

• Try the following:

1) What if you remove the parentheses around (count + 1)?
2) What happens if we try this way of breaking a line:
```
System.out.print("Counting...
    up: " + (count + 1));
```
3) How can we get all this output to print all in one line?

• Other examples (textbook): Countdown.java  Facts.java
Escape Sequences

• What if we wanted to print the quote character?
• Let’s try something like this…

    System.out.println ("I said "Hello" to you.");

• An escape sequence is a series of characters that represents a special character
• An escape sequence begins with a backslash character (\)

    System.out.println ("I said \"Hello\" to you.");
# Escape Sequences

- Some Java escape sequences:

<table>
<thead>
<tr>
<th>Escape Sequence</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\b</code></td>
<td>backspace</td>
</tr>
<tr>
<td><code>\t</code></td>
<td>tab</td>
</tr>
<tr>
<td><code>\n</code></td>
<td>newline</td>
</tr>
<tr>
<td><code>\r</code></td>
<td>carriage return</td>
</tr>
<tr>
<td><code>\&quot;</code></td>
<td>double quote</td>
</tr>
<tr>
<td><code>'</code></td>
<td>single quote</td>
</tr>
<tr>
<td><code>\</code></td>
<td>backslash</td>
</tr>
</tbody>
</table>
public class Roses {
    public static void main (String[] args) {
        System.out.println("Roses are red, \n\tViolets are blue, \n" +
        "Sugar is sweet, \n\tBut I have "commitment issues\", \n\t" +
        "So I'd rather just be friends\n\tAt this point in our " +
        "relationship.");
    }
}
Quick Check

Write a single `println` statement that produces the following output:

"Thank you all for coming to my home tonight," he said mysteriously.
Next: variables

From Lab 1:

```java
int x = 42, count = 100;
String name = "Kripke";

System.out.println ("Howdy " + name);
System.out.println ("The answer is " + x);
```
int x = 42, count = 100;
String name = "Kripke";

System.out.println ("Howdy " + name);
System.out.println ("The answer is " + x);

name = "Sheldon";
x = 33;

System.out.println ("Howdy " + name);
System.out.println ("The answer is " + x);
Variables

- A variable is a name for a location in memory.
- A variable must be declared by specifying the variable's name and the type of information that it will hold.

```java
int sum;
double milesPerGallon;
String name, petName;
```
## Some types of data in Java

<table>
<thead>
<tr>
<th>type</th>
<th>set of values</th>
<th>literal values</th>
<th>operations</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>char</code></td>
<td>characters</td>
<td>'A' '@'</td>
<td>compare</td>
</tr>
<tr>
<td>String</td>
<td>sequences of characters</td>
<td>&quot;Hello World&quot;</td>
<td>concatenate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;jackie123&quot;</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>integers</td>
<td>17 12345</td>
<td>add, subtract, multiply, divide</td>
</tr>
<tr>
<td>double</td>
<td>floating-point numbers</td>
<td>3.1415 6.022e23</td>
<td>add, subtract, multiply, divide</td>
</tr>
<tr>
<td>boolean</td>
<td>truth values</td>
<td>true false</td>
<td>and, or, not</td>
</tr>
</tbody>
</table>

Java Bootcamp      Dr. Papalaskari      Villanova University