Constants, Variables, and Data Types

Mrs. Tillman
Java Programming
Basic Java Syntax and Semantics

Data Types:
- Primitive data types
  - Numbers (integer and floating point)
  - Characters (“A”, “B”, etc.) (char)
  - Booleans (true and false) (boolean)
  - Numbers use operators (addition and multiplication)
- Objects
  - Are sent messages
  - Must be instantiated before use
- Strings
  - Are objects
  - Are sent messages
  - Do not need to be instantiated
  - Can be combined using the concatenation operator (+)
Basic Java Syntax and Semantics

Numeric Data Types:

- 6 numeric data types are used in Java:
  - int (integer, no decimals)
  - double (double precision floating-point numbers; i.e. numbers with decimals)
  - short
  - long
  - byte
  - float

Not in AP CS subset.
**Basic Java Syntax and Semantics**

Some Java numeric data types:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>STORAGE REQUIREMENTS</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>4 bytes</td>
<td>-2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>double</td>
<td>8 bytes</td>
<td>-1.79769313486231570E+308 to 1.79769313486231570E+308</td>
</tr>
</tbody>
</table>

I hope I don’t have to memorize this.
Basic Java Syntax and Semantics

**Literals:**

- By literal, we mean any number, text, or other information that represents a specific value and does NOT change:
  - $\pi$ (PI)
  - 7.21
  - “Hello World”

- In the statement:
  - `int month = 10;`
  - 10 is the literal
Basic Java Syntax and Semantics

Identifiers

- An identifier is the name of a variable, method, or class.
- Java imposes the following rules on identifiers:
  - Can be made up of letters, digits, underscore (_), and dollar sign ($) characters.
  - Cannot start with a digit: `score1` is legal but not `1score`.
  - Cannot use reserved symbols such as `!` and `%`: `money!` is not a legal identifier.
  - Spaces are not permitted: `myScore` is legal but not `my Score`.
  - Cannot use reserved words such as `new`, `public`, `while`. `int newNum = 2;` is legal but `int new = 2;` is not valid.
  - Identifiers are case sensitive: `myScore` and `myscore` are different identifiers.
Basic Java Syntax and Semantics

Questions/Practice:

- What is the data type of each of the following values?
  - 56: int
  - "56": String
  - 56.0: double

- Tell me whether each of these identifiers are legal or illegal:
  - 100Answers: illegal
  - void: illegal
  - my_cash: legal
  - <myVar>: illegal
  - money$value: legal
  - lucky number: illegal

- Create a statement to store your phone number and use camel case.
Basic Java Syntax and Semantics

Declarations

- Variables
  - A **variable** is a storage location in memory that has a **type, name (identifier), and value**.
  - Before using a variable for the first time, the program must declare its type.
  - Declare a variable in a **variable declaration statement** (i.e. `int year;`)
  - Several variables can be declared in a single declaration.
  - Remember to use camelCase for variable names!
Basic Java
Syntax and Semantics

Declarations

- Variables
  - Syntax:
    - `<typeName> <variableName> = <value>;` OR
    - `<typeName> <variableName>;
  - Initial values can be assigned in the same statement as its variable declaration:
    - `int x, y, z = 7;`  
      is the same as: `int x;
      int y;
      int z = 7;`
    - `double p, q = 1.41, t;`
    - `String name = “AP Computer Science”;
    - `UrRobot karel = new UrRobot(1,1,East,0);`
    - `String name = 13;`  
      Types do not match
Basic Java Syntax and Semantics

Objects
- Declare the object variable, instantiate or create an object, and assign the object to the variable.
  - `<className> <variableName> = new <className>()`
  - `Robot bot = new Robot(1, 1, North, 0);`

Constants
- The value cannot change
  - `final double SALES_TAX_RATE = 7.00;`
- `final` indicates a variable is declared as a `constant`
- The naming convention for constants is that they are written in UPPERCASE with underlines between words.
- Trying to change the value of a constant after it is initialized will be flagged by the compiler as an error.
Basic Java
Syntax and Semantics

Assignment Statements

- The assignment operator is =
- An assignment statement has the following form:
  `<variableName> = <expression>;`
- The value of the expression on the right is assigned to the variable on the left. It simply replaces the value of the variable:
  ```java
double celsius;
double fahrenheit = 82.5; //Assign 82.5 to variable fahrenheit
celsius = (fahrenheit – 32.0) * 5.0 / 9.0;
```
Assignment Statements

- It is an error to use a variable that has never had anything assigned to it:
  ```java
  int myNumber;
  System.out.println(myNumber);
  ```

- The solution to this problem is:
  ```java
  int myNumber;
  myNumber = 20;  // Assign a value to myNumber
  System.out.println(myNumber);
  ```
Pseudo Code

• Step by step to what you need to do.
• You can keep them in the program if you wish.
• Usually start with all comments
Pseudo Code – please following this order in your coding….or else!

//Create a Scanner Object (if needed)
//Define Variables
//Get input data
//Calculations/Methods
//Display results
Quick Demo

Create a program that declares and prints your name + “you are the greatest!”
Quick Demo

Create a program that declares a person’s name and says their age.