Variables

Somewhere to store information

Overview

- Somewhere to store information
  - Name and Type
- Statements
  - Declaration (creation)
  - Assignment (give a value to a variable)
    - expressions
- Example Program
  - Read input from a user
  - Store it in a variable
  - Perform a calculation

Basics

2. Name

3. Type

1. Somewhere to store information

Basics

- Somewhere to store information
  - Java (and most other programming languages), uses variables to hold information
- Name
  - To access the information
- Type
  - Type of information (e.g. number, Robot, String, boolean value)
  - int (short for integer) can store a whole number.

Statements

- Declaration Statement
  - Creates the variable, specifies the name and the type and reserves the required memory.
- Assignment Statement
  - Place information into the variable
  - Expressions

Example Program

- We want a program that will do the following:

Please type the temperature (deg C): 20
20 deg C is 68.0 deg F

user types this

computer then calculates this

Example Program

```
public class Temperature
{
   public static void main(String[] args)
   {
      int temperature; // Temperature in Celsius

      System.out.print("Enter temperature (deg C): ");
      temperature = Console.readInt();
      System.out.print(temperature);
      System.out.print(" deg C is ");
      System.out.print(9.0 * temperature/5.0 + 32.0);
      System.out.println(" deg F");
   }
}
```

Temperature.java

multiplication symbol

Example Program

- temperature is a variable
- The *declaration* of the variable
  `int temperature;`
  saves a space for a value to be stored later, but doesn't store anything there.
- The *assignment* to the variable
  `temperature = Console.readInt();`
  actually puts something in the space that was created by the declaration

Example Program

```
int temperature;
```

`int` is short for integer and it specifies the integer type. An int is used to represent whole numbers (e.g. 5, 37, -6, 0).

Variable Definition

```
public class Temperature
{
   public static void main(String[] args)
   {
      int temperature;
```

temperature

You don't know what's in there;
temperature can't be used

Variable declaration

Java Syntax: Variable declaration

TypeName variableName;

TypeName variableName = expression;

Examples

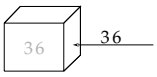
```
double total;
int pennies = 8;
```

Purpose

To define a new variable of a particular type and optionally supply an initial value

Simple Variable Assignment

```
temperature = 36;
```



temperature

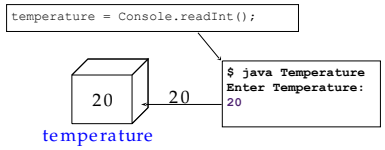
Puts the integer 36 into temperature; now you can use temperature

How input works

```
public class Temperature
{
   public static void main(String[] args)
   {
      int temperature; // Temperature in Celsius

      System.out.print("Enter temperature (deg C): ");
      temperature = Console.readInt();
      System.out.print(temperature);
      System.out.print(" deg C is ");
      System.out.print(9.0 * temperature/5.0 + 32.0);
      System.out.println(" deg F");
   }
}
```


Variable Assignment



The Console class has a **`readInt()`** method which waits for the user to type an integer. This is then assigned to the temperature variable.

The Assignment statement

Java Syntax: The assignment statement

variableName = expression;

Example

```
temperature = Console.readInt();
pennies = 8;
total = pennies + 0.01;
```

Purpose

To assign a new value to a variable

= is not equals!

- Note that the assignment statement

  `temperature = 36;`

  is not saying that temperature is equal to 36, it is an instruction: "place the value 36 into the temperature variable".
- In Java, when you see the **=** symbol, it is not the mathematical equals sign, it is the assignment operator.

Expressions

- An Expression produces a *value* that can be used in Java statements
  - Made out of literals, variables, symbolic constants, and operations
  - Every expression also has a type
  - `3 + 4 + temperature` (has type int)
  - `(3.0 / 4.0) * 70.0` (has type double)
- Can also be calls to a method e.g.

  Console.readInt( )

Type of an Expression

No matter how simple or complicated an expression is, it always has a Java type, just like a variable.

What can you do with an expression?

- An *expression* produces a value, which is then often used in an assignment *statement*
- An expression is on the right hand side of the assignment statement:

```
temperature = 36;
temperature = (x / 7) + y;
temperature = Console.readInt( );
```

Types must match

Changing Variables

- If a variable contains a value and you assign a new value to it, then it overwrites the old value. Consider the two assignment statements

```
temperature = 7;
temperature = 23;
```

The first gives temperature a value of 7. The second one gives it a value of 23. Note that the original value is gone.

The Order of Assignment

- In general, the computer evaluates the expression on the right side, and places that value in the variable on the left side, replacing the old value. So:

```
temperature = temperature + 10;
```

is completely legal; it adds 10 to the value of temperature

Summary

- Somewhere to store information
  - Name and Type (e.g. int)
- Statements
  - Declaration (creation)
  - Assignment (give a value to a variable)
    - expressions
- Example Program
  - Read input from a user
  - Store it in a variable
  - Perform a calculation

Questions