

Notes

CSC140 Foundations of Computer Science

09 February 2015

1. A **package** is a collection of related **classes**.
2. There are two kinds of **classes**:
 - (a) A **class** can be a blueprint for the construction of objects. Each object is an instance of the class. Each object is a collection of related data and methods for operating on that data. Objects have a state that is defined by the values it holds. For example, each instance of the `Color` class stores the amount of red, green, and blue that together produce a particular color. Different instances of the same class may have different states—from the `Color` class it is possible to construct many distinct `Color` objects. The instances of many **classes** have changeable state. For example, methods of the `Color` class can be used to make a color brighter or darker. These operations change the amount of red, green, and blue in the color, and so change the object's state.

To create an instance of this kind of **class** type something like this:

```
Color peach = new Color( 248, 216, 132 );
```

To invoke a method of this kind of **class** type something like this:

```
int red = peach.getRed();
```

You see here the name of an instance of the **class** (`peach`), a period, the name of a method (`getRed`), and parentheses enclosing the arguments for the method (in this case, there are no arguments and so the parentheses are empty).

- (b) A **class** can be a collection of related methods and constants. It is not possible to create instances of this kind of class. Because it is not possible to create objects, it is not possible to create objects that have changeable state. The `Math` **class** is an example of this kind of **class**. Its methods are mathematical functions: sine, cosine, logarithm, square root, and so on. Its constants are $\pi = 3.14159265\dots$ and $e = 2.7182818\dots$

It does not make sense to construct different `Math` objects, because the definitions of sine and π are the same for everyone.

To invoke a method of this kind of **class** type something like this:

```
double squareRootOf2 = Math.sqrt( 2.0 );
```

You see here the name of the **class** (Math), followed by a period, followed by the name of the method (sqrt), followed by the arguments for the method enclosed in parentheses (in this case, single argument whose value is 2.0).

- (c) A method is a sequence of arithmetic and logical operations with a single, well-defined purpose.

Here are the steps for writing a method:

- i. Describe the method's purpose in a single, plainly worded, concise sentence.
 - ii. Give the method a name that signifies its purpose.
 - iii. Specify what kind of result the method will return to its caller.
 - iv. Specify what kind of information the methods must receive from its caller—that is, specify the number and type of the arguments.
 - v. Write sequence of Java statements that direct the computer to carry out arithmetic and logical operations.
- (d) A variable is a named location in the computer's memory. Six attributes completely describe a variable:
- i. name
 - ii. location (address)
 - iii. type
 - iv. value
 - v. scope
 - vi. lifetime
- (e) Variables, arguments of methods, the values of expressions, and the values returned by methods all have types. The type tells us...
- i. how the value is encoded in the computer's memory— the same binary number (pattern of ones and zeroes) can represent different values for different types
 - ii. what kinds of operations are permitted