# Methods in Java

# Method Overloading

# Passing Parameters

The phrase “passing parameters” is used to describe the input and output that occurs with methods.

The parameters are the input values we pass to the procedure when we call it.

For example:

```
doSomething(input1, input2, input3);
```

The parameters are input1, input2, and input3.

# Parameters Do Not Change

In some languages, the parameter can be changed by the method (e.g., Turing, C, C++).

In Java, the parameter cannot be changed by the method.

Whenever a parameter is passed to a method in Java, the method makes its own copy of the data. Any changes only affect the copy, not the original.

# Same Method, Different Parameters

In some cases, we may want a method to accept a wide variety of parameters (including none at all).

In the simplest case, the purpose of the method may be essentially the same, but acting on different data types.

More complicated, but still valid, would be methods having the same theme, but the methods themselves may vary significantly.

# Overloading Methods
## (for the same purpose)

Consider the following method definitions. The data types change, and the code would differ slightly, but the goal is always the same.

```
public static int max(int a, int b)
public static double max(double a, double b)
public static char max(char a, char b)
public static String max(String a, String b)
```

# Overloading Methods
## (for the same theme)

In this case, as the input parameters change, the code for each method will also start to change more significantly. Some methods may even call different versions of their own name.

```
public static void rollDie()
  // output result of single 6-sided die
public static void rollDie(int numRolls)
  // output result of N 6-sided rolls
public static void rollDie(int numRolls, int numFaces)
  // output result of N, M-sided rolls
```

# Overloading Methods
## (existing examples in Java)

Overloaded methods are commonly used in Java. Most examples found will be for changes in code required across different data types. For example, the max method already exists in the Java Math library:

```
public static int max(int a, int b)
public static long max(long a, long b)
public static double max(double a, double b)
public static float max(float a, float b)
```