

# Procedures to Simplify Programs

A procedure is very general. In fact, a procedure can be written to duplicate the effect of almost any block of code.

One of the best reasons to use procedures is to simplify our main program by isolating complicated code as procedures (and sometimes functions, where appropriate).

# Procedures to Simplify Programs

By using procedures (with good names) in our main program, it is much simpler to understand.

The procedures hide some of the complicated details, which is called encapsulation.

```
// Main Program
```

```
initDisplay()
```

```
validateUser()
```

```
mathQuiz()
```

```
shutdown()
```

# Passing Parameters

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

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.

# Example – Adding Parameters

```
// print a line of '*' characters
class DemoParameters
{
    public static void main(String[] args)
    {
        for (int count = 0; count<10; count++)
        {
            System.out.print("*");
        }
        System.out.println();
    }
}
```

# Example – Adding Parameters

```
// print a line of '*' characters
class DemoParameters
{
    public static void printRow()
    {
        for (int count = 0; count<10; count++)
        {
            System.out.print('*');
        }
        System.out.println();
    }

    public static void main(String[] args)
    {
        printRow();
    }
}
```

# Example – Adding Parameters

```
// print a line of '*' characters
class DemoParameters
{
    public static void printRow(int numCh)
    {
        for (int count = 0; count < numCh; count++)
        {
            System.out.print('*');
        }
        System.out.println();
    }

    public static void main(String[] args)
    {
        printRow(15);
    }
}
```

# Example – Adding Parameters

```
// print a line of '*' characters
class DemoParameters
{
    public static void printRow(char ch, int numCh)
    {
        for (int count = 0; count < numCh; count++)
        {
            System.out.print(ch);
        }
        System.out.println();
    }

    public static void main(String[] args)
    {
        printRow('*', 15);
    }
}
```