

ANATOMY OF A PROGRAM

*Leslie Archibald under the direction of Mrs. Debbie Sorrentino
Niagara County Community College, Honors Institute*

VARIABLES

Variables store information that the program manipulates while it is running. There are multiple data types, and each one holds a different type of data within them. Variables must be *declared* in a program before they can be used. All variables must have been *initialized* and used by the time the program has finished running.

In the non-digital world, a comparable item would be a storage bin. These bins are filled with objects, and have labels that describe their contents so a layperson knows what’s inside without having to open the bin up. Similarly, variables hold information, and are given a name so the human programmers looking at the code better understand what data the variable holds. For example, a double variable that holds the cost of an item could be declared as double price

Variables can be used for many things. The list is long, but some of the most basic uses include:

- Mathematical computations
- Comparisons
- Passing values into a function
- Holding values for later use
- Creating a list of related values in arrays

There are other uses as well, but these are some of the more common ones. Without variables it would be impossible to create the programs we use in everyday lift.

```

1 import java.util.Random; // imports Random, a predefined class
2
3 public class PosterClass { // creates PosterClass
4
5     public static void main(String[] args) { // Main function
6
7         // Common Variable Types
8
9         int i = 1, inum = 3; // holds positive or negative whole numbers
10        double d = 3.5, dnum = -4.3; // holds positive or negative decimal numbers
11        char c = 'a'; // holds a single alphanumeric character
12        String s = "Hello Everyone!"; // holds lines of alphanumeric characters
13
14        Random rngGenerator = new Random(); // creates a class "variable", an object or instance
15        int[] scores = {90, 85, 60, 99, 73}; // holds multiple variables of the same data type
16
17        inum = i - inum; //performs 1 - (-3) and stores it in inum
18        dnum = d + dnum; //performs 3.5 + (-4.3) and stores it in dnum
19
20        System.out.println("Your stored character is: " + c); // prints the contents of c to a screen
21        System.out.println("Your stored message is: " + s); // prints the contents of s to a screen
22        System.out.println(scores[3]); // prints the element stored at index 3 of the array
23
24
25        PosterClass example = new PosterClass(); // declares object using the constructor function
26
27        // function call to rng class
28        i = rngGenerator.nextInt(99) + 1; // generates a random number between 0-99 and adds 1 to it
29        // this value then gets stored in i, overwrites the previous value
30
31        example.displayMessage(s); // calls the display function and executes the code within it
32
33
34        i = example.performAddition(inum, i); // Here we call the two mathematical functions, and then print out the results
35        d = example.performMultiplication(dnum, d);
36
37        System.out.println("The sum is: " + i + " and the product is: " + d);
38
39    }
40
41    // display function prints a message to the screen
42    void displayMessage(String message) {
43
44        System.out.println(message);
45    }
46
47
48    // Addition function adds together two values, and then returns the resulting sum
49    int performAddition(int num1, int num2) {
50
51        int sum; // declare an integer variable
52        sum = 0; // initialize the variable to zero
53
54        sum = num1 + num2; // add the numbers that are passed into the function
55
56        return sum; // send the answer back to the main function
57    }
58
59
60    // Multiplication function multiplies two numbers, and then returns the resulting product
61    double performMultiplication(double num1, double num2) {
62
63        double product; // declare a double variable
64        product = 0; // initialize the variable to zero
65
66        product = num1 * num2; // multiply the numbers passed into the function
67
68        return product; // send the answer back to the main function
69    }
70
71
72 }
73
74
```

FUNCTIONS

Functions are sections of a program that perform a specific task. They are given a name that describes their purpose to help the program users understand at a glance what the function is supposed to do.

There are a few types of functions:

- Main functions– the code that runs automatically when a program is started
 - sometimes referred to as the main program
 - This is a special function and cannot be called by other functions
- Set functions – these set the values of private variables
- Get functions – these get the values of private variables
- Void functions – functions that do not return a value. They are commonly used for displaying things
- Value returning functions - functions that send a data value back to the main program for use

The main advantages of functions is that they are reusable. You can call them as many times as you want, whenever you want. Instead of having to rewrite the same code over and over, a programmer simply has to put that code into a function. They then can call that function as many times as necessary in order to use the code within it.