**Java Common Concepts**

# Argument

A value that is passed to a Parameter.

# Class

A “blueprint” for an object. Programmers write most of their code inside classes.

# Constructor

A special method that runs one time when an object is first created. The constructor must have the same name as the class and must not have a return type or void. Every class has a constructor – if the programmer does not define one, the compiler will create an empty one.

# Function

A method that returns a value to its caller. For example, a function that adds two numbers would return the result. A function must define the type of data to be returned.

# Inheritance

A system by which an object is based on another object. The “child” object receives (inherits) all of the “parent” object’s functionality, but can also change or add to that functionality.

# Method

Something that an object can do, such as add numbers. There are two types of methods – subroutines and functions.

# Object

1. A discreet “thing” in a computer’s memory that has both properties and methods.
2. The base from which all other types inherit.

# Parameter

A value that is required by a method. A caller must pass a value (argument) to that Parameter when calling the method.

# Private

A keyword that makes a programming element only available within its container.

# Property

A piece of data that an object can hold. In Java, properties are simply public variables.

# Public

A keyword that makes a programming element available to any caller.

# Subroutine

A method that does not return a value to its caller. A subroutine is defined by the “void” keyword in place of a returned data type.

# Type

A definition of an object. Every class that you write creates a new Type, and objects of this Type can now be created. Java also has several built-in types, such as *int* and *String*.

# Variable

A named pointer to an in-memory object. It can be useful to think of a variable as a “box” that you can place a value in.