**Variables**

* Variables are used to store a information in a program and it can also be manipulated in a computer program.
* Think of variables as containers that hold information.
* Variables sole purpose is to label and store data in memory.

**Data Type**

* A Data Type in the world of programming is a classification which specifies which type of values a variable has.
* And also a Data Type defines what type of mathematical, relational, or logical operations can be applied to it without an error.
* For Example, a String can be used to declare a text and an integer can be used to declare a number.
* In the world of Programming there are two types of Data Types:

**Data Types**

**User-defined data types**

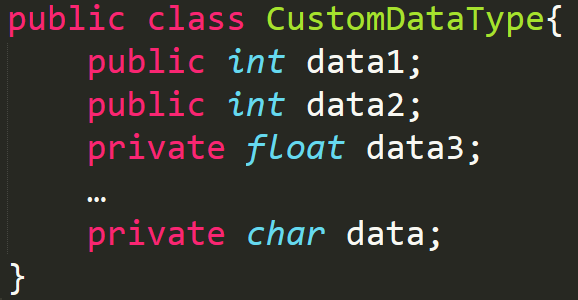
**System-defined Data Types**

**System-Defined Data Types**

* Data types that are defined by the system are called ***primitive*** data types.
* The primitive data type provided by many programming languages are: **int, float, char, double, bool, etc.**

**User-Defined Data Types**

* There will be a situation where primitive data types will not be enough for your program. So, most programming languages allow the users to define their own data types called as user-defined data types.
* Good example of user-defined data types are: structures in C/C++ and classes in java.
* Example:

****

**Data Structures**

* Data Structure is a particular way of storing and organizing data in a computer so that it can be used efficiently.
* General data structure types include the array, linked lists, stacks, queues, trees and so on.
* Any data Structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways.

**Abstract Data Structures**

* The system provides data types (int, float, etc.) support basic operations such as addition and subtraction. The system provides implementations for the primitive data types. And For user-defined data types we also need to define operations. The implementation for these operations can be done when we want to use them. That means user-defined data types are defined along with their operations.
* To simplify the process of solving the problems, we combine the data structures along with their operations and call it as ***Abstract Data Types (ADTs)***
* An ADTs consists of two parts:

1. Declaration of data
2. Declaration of operations.

* You can take Java's List interface as an example. The interface doesn't explicitly define any behavior at all because there is no concrete List class. The interface only defines a set of methods that other classes (e.g. ArrayList and LinkedList) must implement in order to be considered a List.
* A **collection** is another abstract data type. In the case of Java's Collection interface, it's even more abstract than List.
* While defining the ADTs do not worry about the implementation details. They come into picture only when we want to use them. Different kinds of ADTs are suited to different kinds of applications, and some are highly specialized to specific tasks.