Chapter 1 of Shaffer’s (2011) textbook, defines several concepts that we will need to understand to succeed in this course. Below, I summarize these concepts for the novice as well as describe how each concept relates to the others.

* Type - A type is all the values comprising a set or collection. Something to keep in mind, is that a type is a theoretical category, not a software implementation.
* Data Item - A Data Item is a specific value that is a member of the theoretical Type.
* Data Type - A data type is comprised of all of the operations that can be performed on a Type in addition to the set of values that make up the Type itself. For example, the type *integer* can be added, subtracted, multiplied, and divided. Therefore, the set of integers in addition to the four mathematical operations comprises the Data Type.
* Abstract Data Type - In contrast to the theoretical constructs above, an Abstract Data Type (or ADT, for short) is the Data Type as realized in software. In other words, the theoretical Type and the operators which can manipulate it are conceptualized in a program. The exact implementation of how the operators perform the manipulations does not matter for the ADT. All that matters is that the type and its operators are collectively added to software and that the ADT’s interface is defined by the type and operators that manipulate it.
* Data Structure - The Data Structure comprises *how* specifically the Abstract Data Type is implemented in software.
* Class - The Class is the ADT and its Data Structure as implemented in software. The class, then is analogous to the Data Type, but is implemented in software.
* Member Function - A Member Function is the exact method used to perform an operation of a class. That is, a member function is the code which manipulates the Type in software.
* Data Members - Data Members are the variables, as coded in software, that define the memory space of Data Items.

Remember that Types, Data items which comprise Types, and the Data Type which encompases the Type and it’s operators, are theoretical constructs, not software implementations (Shaffer, 2011). Conversely, ADTs, Data Structures, Classes, Member Functions, and Data Members are specific to the programming language being used.

Reference

Shaffer, C. A. (2011). A Practical Introduction to Data Structures and Algorithms Analysis (Ed 3.1). Retrieved from <http://people.cs.vt.edu/~shaffer/Book/Java3e20110103.pdf>