122COM

Introduction to Algorithms

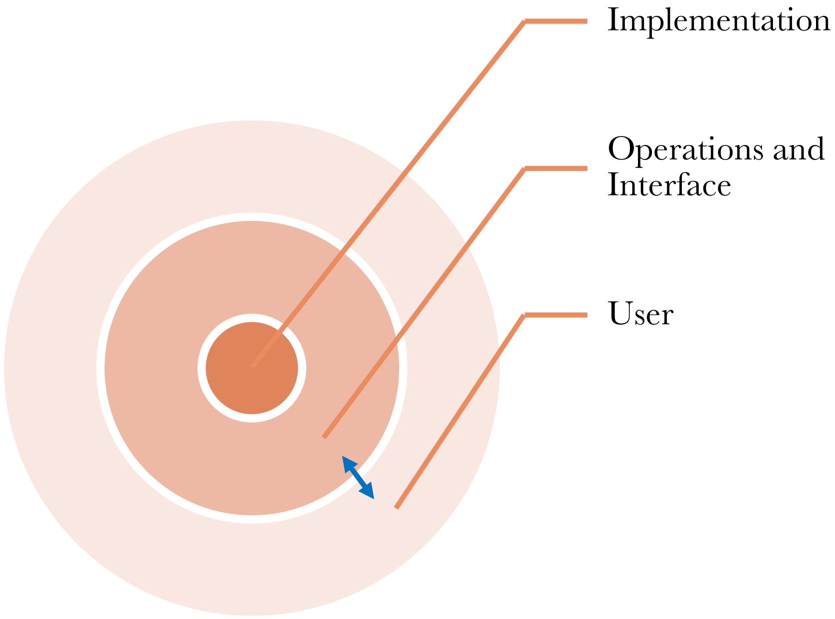
Task 4 – Abstract Data Types

Student Name: **Chalumuri Venkata Nag Rithin Naidu**

Student ID: **7030330**

Abstract Data type is defined in terms of its data items and associated operations, not how it is implemented. Arrays and Vectors are data structures which focus on the idea of specific implementation details, whereas abstract data types don't. In ADT, focus is on operations and what the data is representing not on how it is implemented. We are basically creating an encapsulation around the data. The idea is that by encapsulating the details of implementation, we are hiding them from the user's point of view. This is called information hiding. Several programing languages support this concept through Classes, additional techniques such as Inheritance, Polymorphism can also be used in modelling ADT.

Here is a picture of what an abstract data type is and how it operates :



We can consider the example of, Set Data type which can be created using classes. What we need to think about when modelling an ADT for a set is the data (size of set, items in set, type of set) and the operations such as adding elements, checking if an element is in the set, removing elements, Union, Intersection, and Difference of two sets. The user gets to interact and perform these operations. The abstract data type is the shell that the user interacts with. The implementation is hidden one level deeper (innermost shell). The user is not concerned with the details of the implementation.

When we are talking about Implementation it gets into the topic of data structures such arrays, linked lists, etc. However, we should note that Internals of an ADT can vary widely between implementations. One the main uses of ADTs is that, without changing the way the user interacts with data, the programmer gets the flexibility of implementing different techniques to perform underlying operations. When dealing with large scale problems, ADT allows programmers to see the "big" picture and overall functionality of the program without getting lost in the small details such as implementations. This is a big advantage when modelling complex data structures.

References

1. Anon. (2016) *Why Study Data Structures And Abstract Data Types? — Problem Solving With Algorithms And Data Structures* [online] available from <http://interactivepython.org/runestone/static/pythonds/Introduction/WhyStudyDataStructuresandAbstractDataTypes.html> [15 July 2016]

2. Snow, D. (2012) *Data Structures: Abstract Data Type (ADT)* [online] available from <https://www.youtube.com/watch?v=HcxqzYsiJ3k> [15 July 2016]