Abstract Data Types.
Approach with Object-Oriented Programming

David Bradshaw

CSCI 3300

Spring 2011

What is an Abstract Data Types?

- Abstract = Anything that is conceptual.

- Data = Any piece of information that is present.

- Type = Category where something belongs.

From this we can see that Abstract Data Types are:

A conceptual piece of data that belongs to a certain category.

(Dale, 2-3)

Data Abstraction

- Abstraction: Creating a vague way of specifying types.

- Abstractions is seen in many science and business fields.

- For Example; taxonomy is a form of categorizing animals into abstract views to sort among the species.

- Most programmers have used this principle without even knowing it.

(Sebesta, 470)

Abstract Data Types Used in Programming

- Used in most typed languages
- Code to the right is a Java Object-Oriented Program.
- Main Program calls TriangleClass which extends the GeometricObject to compute needed geometric principle

```java
package bradshawlab;
public class TriangleProject {

    public static void main(String[] args) {
        TriangleClass Triangle = new TriangleClass(1, 1, 1);
    }
}


public class GeometricObject {

    private color = "white";
    private boolean filled;
    private java.util.Date dateCreated;
    }


public class TriangleClass extends GeometricObject {

    private double side1 = 1.0;
    private double side2 = 1.0;
    private double side3 = 1.0;

        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }
}
```

Abstract Data Types effect on Object-Oriented Programming

- There are many similarities between ADT and Object-Oriented Programming; for example, they both use abstraction for their core principles.
- Both ADT and Object-Oriented Programming have the same origin of SIMULA 67.
- Both ADT and Object-Oriented Programming use the same principle of pulling data from a different class (or object) to fill in data with what is already known.

(Cook)

Summary

- Abstract Data Types are an amazing concept that makes life easier for programmers.

- Simplistic programming languages are possible because of ADT and Object-Oriented Programming's ability to extend for needed data.

- ADT allow for easy programming and 'behind the scene' operations of type functionality.

(Sebesta, 470), (Cook)

Works Cited

"Abstract Data Type." Encyclopædia Britannica. Encyclopædia Britannica Online. Encyclopædia Britannica, 2011. Web. http://www.britannica.com/EBchecked/topic/1086575/abstract-data-type.

Cook, William. "Object-Oriented Programming Versus Abstract Data Types." REX Workshop/School on the Foundations of Object-Oriented Languages (1990): 151-178. Web. http://www.cs.utexas.edu/~wcook/papers/OOPvsADT/CookOOPvsADT90.pdf.

Dale, Nell, and Henry Walker. Abstract Data Types: Specifications, Implementations, and Applications. 1st Edition. Lexington: D.C. Heath and Company, 1996. Google eBook.

Mitchell, John, and Gordon Plotkin. "Abstract Types have Existential Type." ACM Transactions on Programming Languages and Systems 10.3 (1988). Web. http://theory.stanford.edu/~jcm/papers/mitch-plotkin-88.pdf.

Sebesta, Robert. Concepts of Programming Languages. 8th Edition. New York: Pearson Education, 2008. Print.

"Types, Values, and Variables." Java Language Specification, Third Edition. Sun Microsystems, Inc., 2005. Web. http://java.sun.com/docs/books/jls/third_edition/html/typesValues.html.