

Angela Chen

January 28, 2017

Definitions Assignment

Introduction: The objective of this assignment is to learn how to write the definition of a relatively complex term for an audience that may not fully know the technical meaning of the term. The criteria include choosing a non-technical audience and writing three different types of definitions: parenthetical, sentence, and expanded. Furthermore, four expansion strategies must be used with one of them being visual representation.

Term: Abstraction

Situation and Audience: University students taking CPSC 210 attend lectures on programming in Java. These students do not have a lot of knowledge nor experience in Java programming yet. (Note: Students would already know what classes are by the time that abstraction is introduced.)

Parenthetical Definition: Abstraction (ignoring complex details to focus on the generalized features) is one of the four major concepts in object-oriented programming (Nookala, 2016).

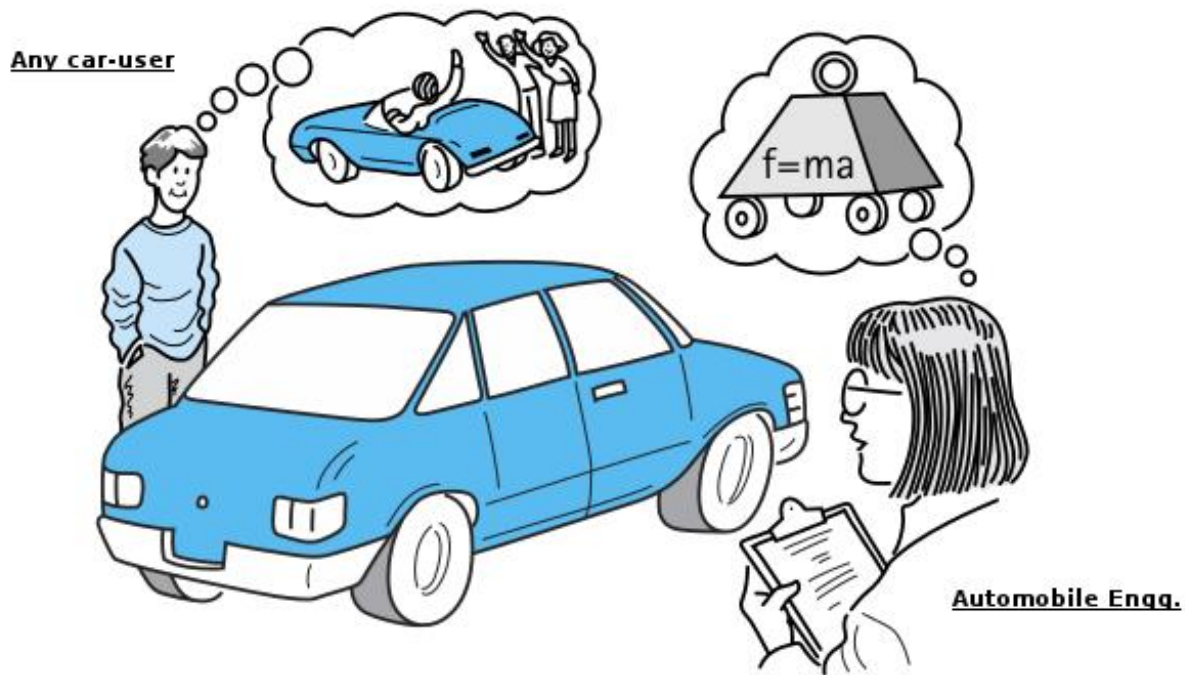
Sentence Definition: In object-oriented programming, abstraction is a mechanism for providing a generalization of key features without including implementation details (Rouse, 2014). A real-life metaphor is when travelling by airplane, one does not need to know how the engine works or how the aircraft was constructed. One can still perform the action of travelling by airplane without knowing all that information.

Expanded Definition:

History – What is its history?

The concept of abstraction in computer science originated from abstract data types (ADT). A typical ADT are integers which are represented by values such as "...-3, -2, -1, 0, 1, 2, 3..." The integer class is an abstraction that hides the complexity of various mathematical operations. ADTs were first proposed by Barbara Liskov and Stephen N. Zilles in 1974 ("Abstract data type", n.d.).

Visual Representation – What does it look like?



An abstraction includes the essential details relative to the perspective of the viewer

Figure 1. A visual representation of abstraction in object-oriented programming. (2014, August 2017). Retrieved January 29, 2017, from <http://www.codingeek.com/oops/object-oriented-programming-best-explanation/>

In Figure 1, an abstraction for any car-user is driving the car. From the man's perspective, he does not need to know how the engine works or know how Newton's three laws of motion is applied to the car, unlike the automobile engineer. For the automobile engineer, Newton's three laws of motion are abstractions—three generalized statements that hide the implementation.

Comparison and Contrast – How does it resemble or differ from something else?

Abstraction and encapsulation are similar concepts but they mean very different things. They both involve the process of hiding details to reduce complexity and improve efficiency (Rouse, 2014). However, in Java abstraction hides the implementation details while encapsulation hides the information (Chauhan, 2009). Consider for instance, adding two numbers together. How are two positive numbers added? How are two negative numbers added? How are a positive and a negative number added? What about adding decimal numbers? In this example, abstraction is hiding *how* different types of numbers are added together and encapsulation is hiding *what* types of numbers are added together. Calling the add function on two numbers should just work

without needing to know or understand the complexity or implementation details of this function.

Examples – How is it used or applied?

An abstract method in a Java class leaves the actual implementation to child classes (Tutorials Point, n.d.); however, the overall purpose of the method remains the same. For example, a payment function can be declared as an abstract method in the CreditCard class. Then child classes like Visa or MasterCard will inherit the abstract payment method from CreditCard and have a different implementation for this method. Maybe the purchase amount is credited to the payer's account before the credit limit is checked, or vice versa. The abstract payment method in CreditCard establishes the purpose of the function regardless of the order of operation in the child classes.

REFERENCES

- Abstract data type. (n.d.). In Wikipedia. Retrieved January 29, 2017, from https://en.wikipedia.org/wiki/Abstract_data_type
- Chauhan, N. (2009, April 13). Difference between abstraction and encapsulation? [Msg 7]. Message posted to <http://stackoverflow.com/questions/742341/difference-between-abstraction-and-encapsulation>
- Nookala, S. (2016, April 14). *Abstraction in Java*. Retrieved January 28, 2017, from <http://java.meritcampus.com/core-java-topics/java-abstraction-abstraction-in-java>
- Rouse, M. (2014, June). *Abstraction*. Retrieved January 28, 2017, from <http://whatis.techtarget.com/definition/abstraction>
- Tutorials Point. (n.d.). *Java – abstraction*. Retrieved January 29, 2017, from https://www.tutorialspoint.com/java/java_abstraction.htm