

Tutorial 11 – Abstraction

Contents

| | |
|---|----------|
| 1.0 Abstraction and Information Hiding - Definitions | 3 |
| 2.0 Real-Life Examples of Abstraction | 3 |
| 2.1 Example 1: - Daisy the Dog | 3 |
| 2.2 Example 2: - Driver of a Car..... | 4 |
| 2.3 Example 3: - A School with Students | 4 |
| 3.0 Abstraction in the Context of Programming..... | 4 |
| 3.1 Example 1 – How a Casual User Views a Program | 4 |
| 3.2 Example 2 – How a Person who Implements/Programs a Program views a Program | 4 |
| 3.3 Example 3 – How does a Programmer view a Sub-routine/Function/Procedure | 4 |

1.0 Abstraction and Information Hiding - Definitions

(1) **Abstraction and Information Hiding** are two sides of the same coin.

(2) **Information Hiding** is the practice of **hiding details**.

(3) **Abstraction** is the **Result with the Details Hidden**.

(4) An **Abstraction** is a model of a complex system that includes only the details that are essential for the viewer to know, so as to allow focus on the main processes/data (and so ignoring the details the viewer does not need to know).

(5) This allows a programmer to implement a system properly (we shall see what we mean by this later).

2.0 Real-Life Examples of Abstraction

The concept of abstraction is used heavily in real-life as well as in the world of computing.

2.1 Example 1: - Daisy the Dog

Let us take a look at a real-life example of Daisy, the Dog.

Question: How would her owner view her as?

Answer: The owner sees as a household pet with a wagging tail that needs walking and leaves dog hairs everywhere!

Question: How would a vet view her as?

Answer: The vet views Daisy as a mammal with organs, flesh and bones that needs medical care.

Question: How would the farmer view her as?

Answer: Daisy, may be helpful as a working dog, helping to round up the sheep.

So, you can see there are different views, or different forms of abstraction depending on the viewer.

We say there are different abstract representations of the dog.

We say that the owner and the vet use different abstract representations of the dog.

2.2 Example 2: - Driver of a Car

Question: How does the driver of a car view the car?

Answer: A driver does not need to know how a car works to drive it from one point to another. That is, he/she does not need to know how the engine works in details.

The driver only needs to know a few basics about how to interact with the car - i.e how the pedals, knobs and steering wheel work. The driver does not really need to know about the technical details of the car in order to drive it. The driver here does not need to know about the internal workings of the car, just the external workings.

So, we use abstraction to reduce the level of complexity of a system. Or we can say abstraction is an useful tool for reducing the level of complexity of a system.

2.3 Example 3: - A School with Students

Question: Consider a school - what details about students would a school kitchen be interested in?

Answer: The school kitchen would be interested in the name of the student and what food allergies each student has, what food restrictions the student has due to other personal reasons, what sort of portions the students eat, how much food to order for each meal for the students, the times of the lunches of the students.

Question: Consider a school - what details about students would a teacher be interested in?

Answer: The teacher would be interested in the students names and associate them with what subjects they are studying, what year they are in, what study requirements they have if any, etc.

These abstract representations differ in the type of information the kitchen and teacher are interested in.

3.0 Abstraction in the Context of Programming

3.1 Example 1 – How a Casual User Views a Program

Question: How would a casual user of a program/software package, say a person using Word or Excel view the program? (I.e. what is the level or type of abstraction?)

Answer: The casual user only sees how to run the program or use the program, so the only thing they need to know is to how to se Word or Excel. They only see the surface of the program/software package.

3.2 Example 2 – How a Person who Implements/Programs a Program views a Program

Question: How would the person who implements (i.e the programmer) the program view it?

Answer: This person will start with the algorithm and needs to understand it thoroughly in order to implement it/i.e in order to program it. This person must understand the inner workings of the algorithm.

This person is like the vet in the previous example.

Remember, an algorithm is a series of logical steps to solve a problem in a finite amount of time.

3.3 Example 3 – How does a Programmer view a Sub-routine/Function/Procedure

Question: How would a programmer who uses a subroutine/function/procedure/API written by another programmer understand matters?

Answer: The programmer only needs to understand the input and output of the subroutine in order to incorporate the subroutine into their program and they do not need to understand the inner workings/details of the program or the internal details of the program.

