Data abstraction

**Data abstraction** is the reduction of a particular body of data to a simplified representation of the whole.

**Abstraction**, in general, is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics.

**Data abstraction** is the programming process of creating a data type, usually a class, that hides the details of the data representation in order to make the data type easier to work with.

**Data abstraction** involves creating a representation for data that separates the interface from the implementation so a programmer or user only has to understand the interface, the commands to use, and not how the internal structure of the data is represented and/or implemented.

Advantages of Abstraction

1. The programmer does not have to write the low-level code.
2. The programmer does not have to specify all the register/binary-level steps or care about the hardware or instruction set details.
3. Code duplication is avoided and thus programmer does not have to repeat fairly common tasks every time a similar operation is to be performed.
4. It allows internal implementation details to be changed without affecting the users of the abstraction.

Object Oriented Data Abstraction

In object-oriented programming theory, abstraction involves the facility to define objects that represent abstract "actors" that can perform work, report on and change their state, and "communicate" with other objects in the system.

The term encapsulation refers to the hiding of state details, but extending the concept of data type from earlier programming languages to associate behavior most strongly with the data, and standardizing the way that different data types interact, is the beginning of abstraction.