

## 2.3.4.8.2 Definition Data

[Return to Object Data Taxonomy](#)

### Overview

[Return to Top](#)

An Object has the ability to **Define Data** that can be accessed throughout the rest of the Object and depending on the public/private attribution of the definitions outside the Object Definition. The kinds of components that can be defined are:

- **Exceptions**, or faults, are abnormal or unprecedented events occurring during or after the execution of an operation that adversely affects the flow of the operation or the flow of the operations that use the object. Although most runtime systems define some standard exceptions such as Array-out-of-bounds, divide-by-zero, or Runtime, each Data object may define its own exceptions based upon the unique characteristics of the operations. For example, a Thermostat object might have a Too-Cold or Too-Hot exception. It is a runtime error of an undesired result or event affecting normal program flow.
- **Structures** or **Objects** are closely related. The easiest way to think of the difference between **Structures** and **Objects** is that structures usually only define the actual data, and object (i.e., **Class**) defines the data structures as well as operations that can be performed on that data. Object Data is a recursive definition. However, in C++, the main difference between a structure and a class (i.e., object) is the assumption of public/private attribution. Structures assume that all components are by default **public** while classes (i.e., objects) assume that all components are **private**. For example, a Thermostat object might define a TemperatureReading structure that is used to report results.
- **Data Type** are definitions of type information associated with the object. **Data Type** information may be numeric and include ranges, enumerations, scalars or collections. For example, the Thermostat might include a type describing the legal range of numeric values or it might define an enumeration type for Fahrenheit, Celsius, or Kelvin. It might also define a type for an array of temperature readings.

**Note:** These components are not required by any Object and the names used to describe these components can be different depending on the programming language.

### DIDO Specifics

[Return to Top](#)

- ☒ [nick][✓ char, 2022-03-20] ~~Is this page still under construction?~~
- ☒ [char][✓ char, 2022-03-20] ~~New Section -- review~~

From:  
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:  
[https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2\\_views:3\\_taxonomic:4\\_data\\_tax:08\\_objects:03\\_def:start](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2_views:3_taxonomic:4_data_tax:08_objects:03_def:start)

Last update: **2022/05/27 19:42**

