

Introduction to Object-oriented Programming

The Traditional Approach to Programming

A brief discussion of the traditional approach to programming will allow us to see why an alternative approach to programming is useful.

C, Pascal, FORTRAN and BASIC are **procedural languages**, that is, each statement in the language tells the computer to do something. In such languages, a program can be viewed as a list of instructions. When programs get longer, a long list of instructions with no organization becomes unacceptable. Thus, programs are organized into functions and subroutines. This is sometimes loosely referred to as the **structured approach**. The structured approach works quite well until programs get to be very large, at which point it begins to break down.

The Problems with Structured Programming

- In procedural languages, the emphasis is on doing things, i.e. in calling functions and subroutines that perform some action. However, in most cases, the reason for a program's existence is **data**. Data is undervalued in the structured approach and is a second-class citizen.
- In a procedural approach, data must be accessible by the program routines and usually they know how the data is stored. If later the data type is changed, many routines will have to be changed. Moreover, data can easily be corrupted, because it is so accessible (perhaps a global variable).
- Procedural programs are often difficult to design – functions and data structures don't always model the real-world.
- Procedural programs are usually difficult to modify. Often “enhancements” introduce bugs.
- New data types are difficult to implement, especially in such a way that the details are kept hidden. An example is the routines in “string.h”. Everyone knows exactly how the strings are stored. If the null character convention were changed, it would create havoc.

Programming with Objects

In view of the difficulties discussed above, computer scientists have looked for other ways of approaching programming, in particular those that put more emphasis on data. One such approach is called **object-orientated programming** or **OOP**. Before describing OOP program development we should discuss objects briefly. In OOP, the word “object” is used in at least two ways. A **real-world object** can be a person, place or thing, while a **programming object** (or **abstract object**) is a simplified model of a real-world object that models the relevant **attributes** of the real-world object. For example, the attributes of a person are their name, age, sex, weight, etc. In addition to attributes, objects have **behaviors**, which are **actions** that can be taken by the object. These are the things that an object **can do**.

C++ Supports Objects

The C++ language supports objects via the class mechanism. A class is very similar to a C struct, but it has many additional capabilities.