Introduction to Scientific Programming in C++

Ramses van Zon Scott Northrup

SciNet/Compute Canada

March 15, 2011

Outline of the course

1. Introduction
2. C review (+make)
3. Running example
4. C++ as a better C
5. Big C++ (object oriented programming)
6. Important libraries
7. Further reading...

Part I

Introduction

Introduction

Programming strategies

1. Procedural programming
2. Structured programming
3. Object oriented programming

Definition

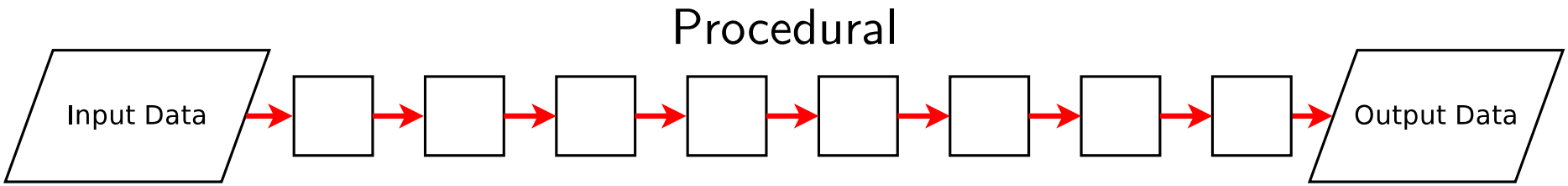
In procedural programming, one takes the view that a sequence of actions are performed on given data.

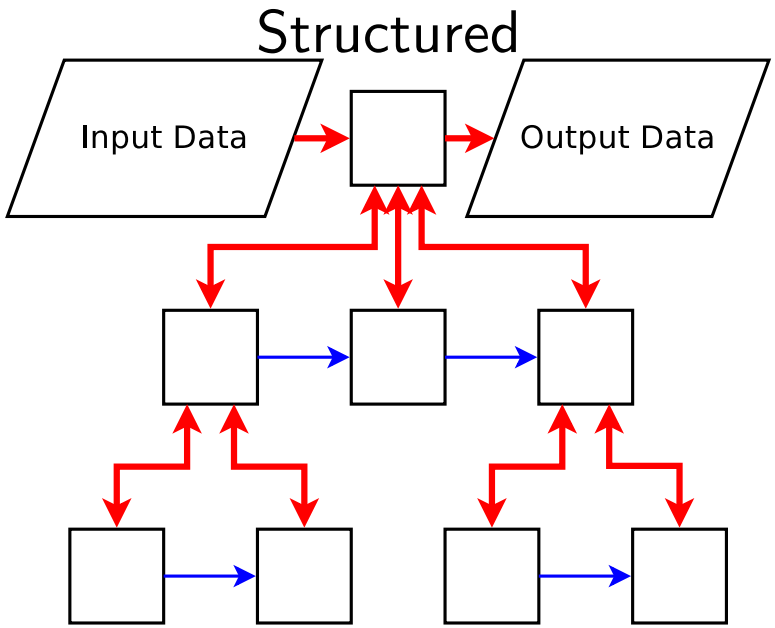
Definition

Structured programming uses a systematic break-down of this sequence of actions down to the simplest task level.

SciNet compute • calcul CANADA

Introduction - Procedural and structured programming





Introduction - Procedural and structured programming

Problems

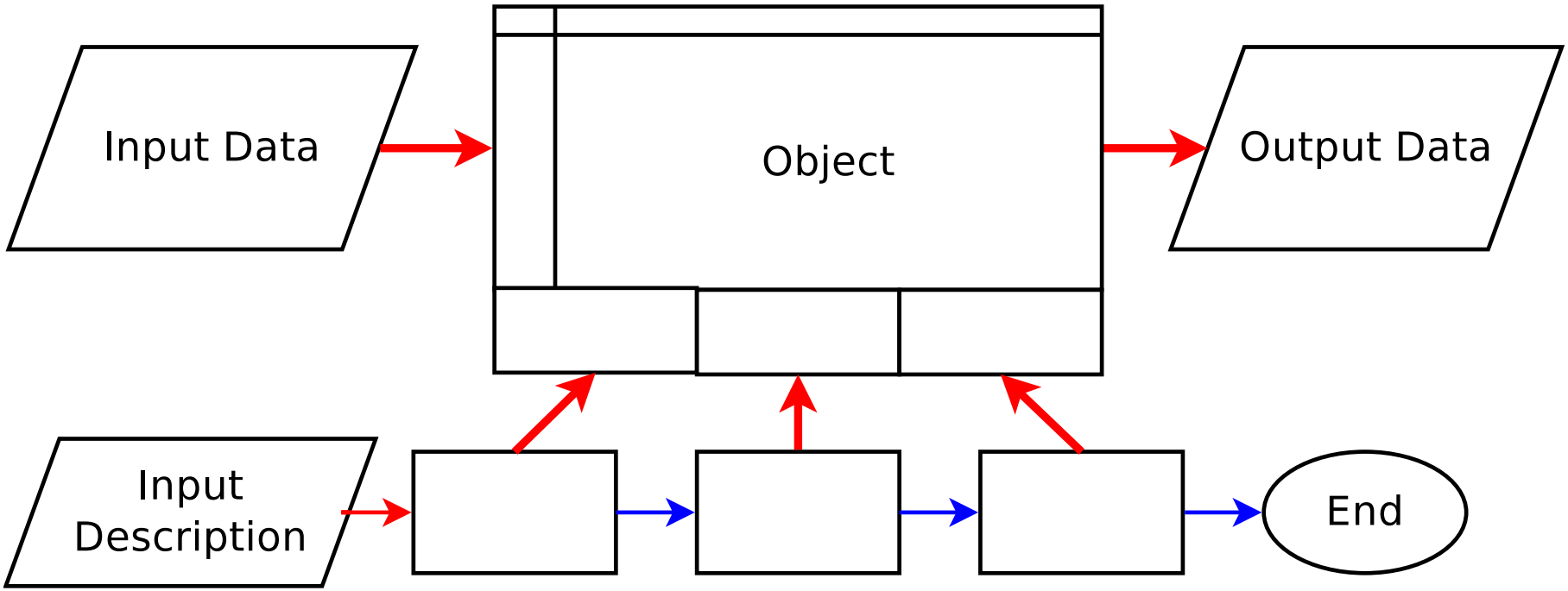
- Complex input data
- Multiple actions to be performed on data
- Separation data+code is bad for reusability
- Leads to reinventing the wheel

One would instead like to build "components" with known properties and known ways to plug them into your program.

Introduction - Object oriented programming

Definition

Object oriented programming treats data and the procedures that act on them as single "objects".



Introduction - Object oriented programming

Advantages

- Complexity can be hidden inside each component.
- Can separate interface from the implementation.
- Allows a clear separation of tasks in a program.
- Reuse of components.
- Same interface can be implemented by different kinds of objects.
- Helps maintanance.

Gotcha: Mind The Cost!

Complexity may be hidden, but you should know:

- the computational cost of the operations
- what temporary objects may be created,
- and how much creating different types of object costs.

On a low level, OOP rules may need to be broken for best performance.

Introduction - Language choice

- You can apply these programming strategies to almost any programming languages, but:
- The amount of work involved in object-oriented or generic programming differs among languages.
- As a result, the extent to which the compiler helps you by forcing you not to make mistakes differs among languages.

C++

- C++ was designed for object oriented and generic programming,
- and C++ has better memory management, stricter type checking, and easier creation of new types than C,
- while you can still optimize at a low level when needed.

SciNet compute • calcul CANADA

Introduction: History of C++

- 1967 *Simula 67:* First object-oriented language by Dahl & Nygaard.
- 1969 – 1973 *C* developed by Dennis Ritchie.
- 1979 *Bjarne Stroustup* writes preprocessor for classes in C
- 1980 Renamed *C with classes*.
- 1983 Now called *C++*.
- 1985 1st edition "The C++ Programming Language"
  C++ supports: *classes, derived classes, public/private, constructors/descructors, friends, inline, default arguments, virtual functions, overloading, references, const, new/delete*
- 1986 Object Pascal (Apple,Borland)
- 1987 pointers to members, protected members
- 1989 multiple inheritance, abstract classes, static member functions
- 1995 ISO/ANSI C Standard
- 1990 templates
- 1993 namespaces, cast operators, bool, mutable, RTTI
- 1995 Sun releases Java
- 1998 ISO C++ standard

source: www.devx.com/specialreports/article/38900