CS315

Joie Ann M. Mac

Reference

- Chapter 12 - Imperative Programming
    - Programming Languages: Principles and Paradigms by Tucker and Noonan

Imperative Programming

- Oldest and most well-developed paradigm
- Mirrors the von Neumann computer architecture
- Typical Languages
  - Fortran
  - Pascal
  - C
  - Clite
  - Ada 83
  - Perl

What Makes Languages Imperative?

- In a von Neumann machine memory holds:
  - Instructions
  - Data
- Intellectual heart: assignment statement
- Other support:
  - Variable declarations
  - Expressions
  - Conditional statements
  - Loops
  - Procedural abstraction

What Makes Languages Imperative?

- Execution of commands is as they appear in memory (sequential)
- Conditional branching and unconditional branching (goto) disrupt normal flow
  - Use of flowchart to model programs
- Turing completeness - a language provides an effective basis for implementing any algorithm that can be designed
  - other paradigms (functional, logic, object-oriented) are also Turing complete

What Makes Languages Imperative?

- An imperative programming language is thus one which is Turing complete and also supports certain common features that have emerged with the evolution of the paradigm:
  - control structures
  - input and output
  - error and exception handling
  - procedural abstraction
  - expressions and assignment
  - library support for data structures

Procedural Abstraction

- **Procedural abstraction** allows the programmer to be concerned mainly with a function interface, ignoring the details of how it is computed.
- The process of **stepwise refinement** (functional decomposition) utilizes procedural abstraction to develop an algorithm starting with a general form and ending with an implementation.
- Ex: `sort(list, len)`

Expressions and Assignment

- **Assignment statement** is fundamental:

```
    target = expression
```

- Copy semantics
  - Expression is evaluated to a value, which is copied to the target; used by imperative languages

Library Support for Data Structures

- There exist vast libraries of functions for most imperative languages
  - avoid reinventing the wheel
- Partially accounts for the longevity of languages like Fortran, Cobol, and C.
- Large collection of classes and functions designed to support the management of complex data structures, I/O functions, exceptions, etc.

C

- Originally designed for and implemented on the UNIX Operating system on the DEC PDP-11, by Dennis Ritchie.
- The operating system, the C compiler, and essentially all UNIX applications programs are written in C.
- Compilers also exist for:
    - IBM System/37-
    - Honeywell 6000
    - Interdata 8/32

C

- Based on BCPL
  - typeless programming language
- Lost its popularity to other languages like C++, Java, Perl, Python
- Has enormous impact on language design both syntactically and semantically

General Characteristics

- Introduced the type cast operation
- Use of braces in place of Algol's `begin` and `end`
- Has:
  - assignment statements
  - statement sequencing
  - if and switch conditional statements
  - while, for and do while loops
  - function calls

General Characteristics

- Data Structures
  - arrays
  - pointers
  - structures (records)
  - union data types
- Lacks the following features
  - iterators
  - exception handling
  - overloading
  - generics

Sample Code: Average

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int ct, number, min, max, sum;
    sum = ct = 0;
    printf("Enter number: ");
    while (scanf("%d", &number) != EOF) {
        if (ct=0) min = max = number;
        ct++;
        sum += number;
        min = number < min? number : min;
        max = number > max? number : max;
        printf("Enter number: ");
    }
    printf(“%d numbers read\n”, ct);
    if (ct>0)
        printf("Average: \t%d\n", sum / ct);
        printf("Maximum:\t%d\n", max);
        printf("Minimum: \t%d\n", min);
    }
    return 0;
}
```

Sample Code: Hello, world!

```
#include <stdio.h>
int main(){
  printf("Hello, world!\n");
  return 0;
}
```

ATTENDANCE

https://forms.gle/CSorAtJCbzSLLpUZ6