M30235 PROGRAMMING APPLICATIONS AND PROGRAMMING LANGUAGES

Lecture 01: Introduction to Programming Languages

Dr Jiacheng Tan

School of Computing
University of Portsmouth

Introduction to Programming Languages

- Domain of uses/applications.
- Categorisations
  - by uses,
  - by paradigms,
  - by ways of task specification.

Programming Domains

- Scientific applications
  - Large numbers of floating-point computations; use of arrays
  - Fortran (**For**mula **Tran**slating System, IBM)
- Business applications
  - Produce reports, use decimal numbers and characters
  - COBOL (**CO**mmon **B**usiness-**O**riented **L**anguage)
- Artificial intelligence
  - Symbols rather than numbers manipulated; use of linked lists
  - LISP (**LIS**t **P**rocessing)
- Systems programming
  - Need efficiency because of continuous use
  - C
- Web Software
  - Eclectic collection of languages: markup (e.g., HTML), scripting (e.g., PHP), general-purpose (e.g., Java)

Language Categories by Uses

- Machine languages: hardware implemented languages. The set of the instructions of a processor.
- Machine code is usually written in hexdecimal numbers. Eg., the instruction of Intel 64 architecture:

```
89 F8 A9 01 00 00 00 75 06 6B C0
03 FF C0 C3 C1 E0 02 83 E8 03 C3
```

Cont'd

- Assembly languages: machine codes are wrapped with alphanumeric symbols so that the instructions are more readable. They also have labeled storage locations, jump targets and subroutine starting addresses, but not much.

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32   PROC          ; procedure begins here
        CMP   AX,97   ; compare AX to 97
        JL    DONE    ; if less, jump to DONE
        CMP   AX,122  ; compare AX to 122
        JG    DONE    ; if greater, jump to DONE
        SUB   AX,32   ; subtract 32 from AX
DONE:   RET           ; return to main program
SUB32   ENDP          ; procedure ends here
```

FIGURE 17. Assembly language

Taken from Google images

Cont'd

- High-level languages: languages that are machine-independent (independent of the machine instructions of any particular processors) and similar to natural language (more readable).
- They are usually characterised with
  - variables, types, subroutines, functions, constants, etc
  - complex expressions (e.g., 2 * (y^5) >= 88 && sqrt(4.8) / 2 % 3 == 9)
  - control structures (conditionals, switches, loops)
  - composite types (arrays, structs) etc.
- Numerous examples, C, Java, …

Cont'd

- System programming languages: differ from application programming languages (which concern solving general application problems such as web authoring, database, or scientific computing) and deal with:
  - memory and process management,
  - I/O operations,
  - device drivers,
  - operating systems.
- Assembly language was used in early days, but C, C++, Ada and etc., are used nowadays.

Cont'd

- Scripting languages are used to write programs in system administration or programming that:
  - analyse or transform a large amount of regular textual information;
  - act as "glue" between different applications (e.g. a web-server and a database);
  - create a simple GUI to control an existing application.
- e.g., Python, PHP, etc.
- Often, such languages
  - are usually interpreted (rather than compiled), and
  - include strong string processing features.

Cont'd

- Domain-specific languages: unlike general purpose languages, they are used in highly special-purpose areas only,
- e.g., PostScript - a language for creating vector graphics for the electronic publishing (Adobe).

Categories by Paradigms

- Procedural
  - A program is built from one or more procedures (also known as subroutines or functions)
  - Central features are variables, assignment statements, and iteration
  - Include languages that support object-oriented programming
  - Include scripting languages
  - Examples: C, Java, Perl, JavaScript, Visual BASIC .NET, C++, …
- Functional
  - Main means of making computations is by applying functions to given parameters
  - Examples: Haskell, LISP, Scheme, ML (*M*eta*L*anguage), F#, etc.
  - Java 8 supports some functional programming.
- Logic
  - Rule-based (rules are specified in no particular order)
  - Example: Prolog

Categories by How Tasks are Specified

- Imperative languages:
  - computing tasks are defined as sequences of commands (statements) for the computer to perform.
  - a program in such language tells computer what to do step-by-step (to make the computer change from one state to the next).
  - e.g., procedural languages.
- Declarative languages
  - in which programs describe their desired results without explicitly listing commands or steps that must be performed.
  - functional and logical programming languages belong to this category.

Next Lecture

- We will give a flavour of different programming languages by implementing a simple algorithm - the Trabb Pardo-Knuth (TPK) algorithm that processes an array.
- Evaluation of programming languages: criteria and performance