Chapter 03

Programming Language

01 Introduction to Programming Language

02 Type of Programming Language

03 Object-Oriented Programming

04 Self Check

05 Practical English

- Computer systems should have programs to provide various services for users. The programs usually are made by programmers. The programmers should have programming languages to make programs. There are various programming languages available and programmers should choose a suitable language for their usage. Then, let’s learn programming languages.

Section 01 Introduction to Programming Language(1)

Programming Language 프로그래밍 언어

- A Programming Language is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. The term programming language usually refers to high-level languages, such as C, C++, Java, Python and so on. Each language has a unique set of keywords and a special syntax for organizing program instructions.
- High-level programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands, called machine languages. Each different type of CPU has its own unique machine language.

Section 01 Introduction to Programming Language(1)

Computer programs

- Also called software
- Are a list of instructions
- Instructions are called code
- CPU performs the instructions

Program execution

- Software executes at the CPU level
- Code to play a sound
  - Code generates an interrupt
  - CPU tells the sound card to play
  - Sound card plays the file
- Programmer creates the code

Section 01 Introduction to Programming Language(1)

■Code

- Statements written in a programming language
- Writing code can be tedious
  - Code must be perfect
  - Order of steps must be exact
- Writing code is quite exciting
  - Problems are solved
  - New ideas are formed

Section 01 Introduction to Programming Language(1)

Machine code

- Recall that computers think in binary
- Code is translated into machine code
  - CPU executes the machine code
- CPUs have a unique machine code

Programming languages

- Simplifies the writing of code
  - English is used to describe the binary
- Original code is called source code
- Several hundred languages exist

Section 01 Introduction to Programming Language(1)

Machine code

- Recall that computers think in binary
- Code is translated into machine code
  - CPU executes the machine code
- CPUs have a unique machine code

Programming languages

- Simplifies the writing of code
  - English is used to describe the binary
- Original code is called source code
- Several hundred languages exist

Section 01 Introduction to Programming Language(1)

Compilers and interpreters

- Converts source code into binary
  - Allows code to execute
- Checks source code for correctness

Compiler

- Creates an executable file
  - Contents are called object code
- Executable can run on its own
- Each language has its own compiler
- C++ and Java are compiled languages

Interpreter

- Runs program one line at a time
- More flexible than compilers
- Slower than compilers
- Always needed to execute program
- Visual Basic and Perl are interpreted

Section 01 Introduction to Programming Language(1)

Structured Programming

- Programming using defined structures
- Creates easy to read code
- Programs are efficient and run fast
- Several defined structures
- Sequence structure
  - Describes the flow of the program
  - Typically executed in order
  - Branching statements allow multiple flows

Section 01 Introduction to Programming Language(1)

Object Oriented Programming

- Enhances structured programming
- Intuitive method of programming
- "objects", which can contain data and code
- All real world items are objects
- OOP develops code versions

- inheritance
- polymorphism
- encapsulation
- message passing
- association

Section 01 Introduction to Programming Language(2)

Assembly Language and Fourth-Generation Language 어셈블리와 4세대 언어

- Lying between machine languages and high-level languages are languages called assembly languages. Assembly languages are similar to machine languages, but they are much easier to program in because they allow a programmer to substitute names for numbers. Machine languages consist of numbers only.
- Lying above high-level languages are languages called fourth-generation languages (usually abbreviated 4GL). 4GLs are far removed from machine languages and represent the class of computer languages closest to human languages.

Section 01 Introduction to Programming Language(2)

Assembly Language and Fourth-Generation Language 어셈블리와 4세대 언어

- Lying between machine languages and high-level languages are languages called assembly languages. Assembly languages are similar to machine languages, but they are much easier to program in because they allow a programmer to substitute names for numbers. Machine languages consist of numbers only.
- Lying above high-level languages are languages called fourth-generation languages (usually abbreviated 4GL). 4GLs are far removed from machine languages and represent the class of computer languages closest to human languages.

Programming Languages

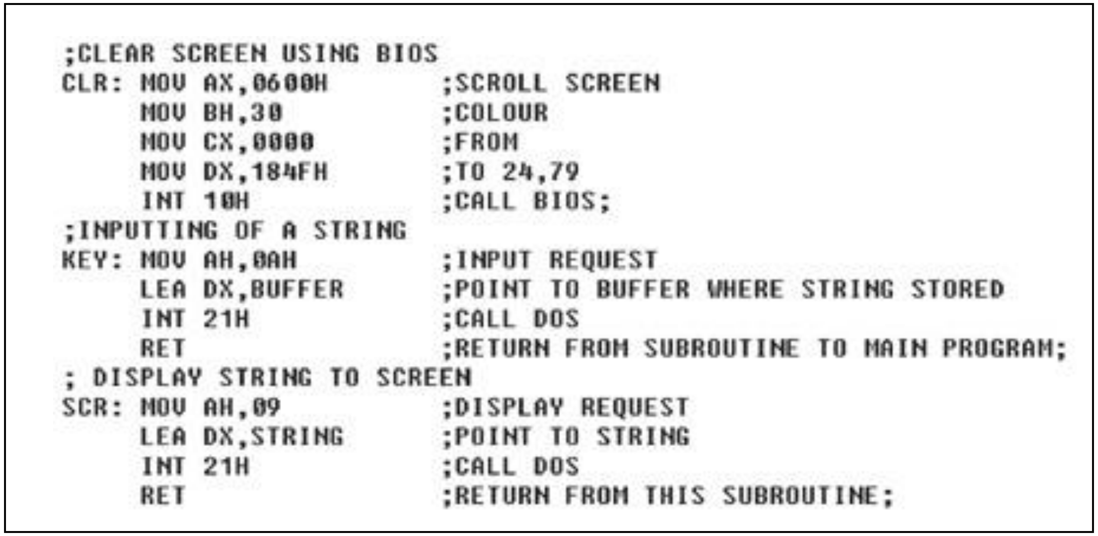
- Used to generate source code
- Avoids using machine code
- Have strict rules of syntax
  - Symbols and punctuation have meaning
  - Spelling must be exact
- Code is converted into machine language

Language Categories

- First generation language
  - Machine languages
  - Written in binary
  - Different for every CPU

Language Categories

- Second generation languages
  - Assembly languages
  - Statements that represent machine code
  - Code converted by an assembler



```
;CLEAR SCREEN USING BIOS
CLR: MOV AX,0600H        ;SCROLL SCREEN
     MOV BH,30           ;COLOUR
     MOV CX,0000         ;FROM
     MOV DX,184FH        ;TO 24,79
     INT 10H             ;CALL BIOS;
;INPUTTING OF A STRING
KEY: MOV AH,0AH          ;INPUT REQUEST
     LEA DX,BUFFER       ;POINT TO BUFFER WHERE STRING STORED
     INT 21H             ;CALL DOS
     RET                 ;RETURN FROM SUBROUTINE TO MAIN PROGRAM;
; DISPLAY STRING TO SCREEN
SCR: MOV AH,09           ;DISPLAY REQUEST
     LEA DX,STRING       ;POINT TO STRING
     INT 21H             ;CALL DOS
     RET                 ;RETURN FROM THIS SUBROUTINE;
```

Language Categories

- Third generation languages (3GL)
  - First higher level language
  - Supports structured and OOP
    - Code is reusable
    - Code is portable
  - C/C++, Java

Language Categories

- Fourth generation languages (4GL)
  - Easier to use than 3GL
  - Coded in a visual IDE
  - Tools reduce the amount of code
  - Object oriented programming
  - Microsoft .Net

Language Categories

- Fifth generation language (5GL)
  - Mystery language
  - May not be created yet
  - Will create software automatically

Systems Development Life Cycle

- SDLC
- Organized way to build programs
- Consists of five phases

Systems Development Life Cycle

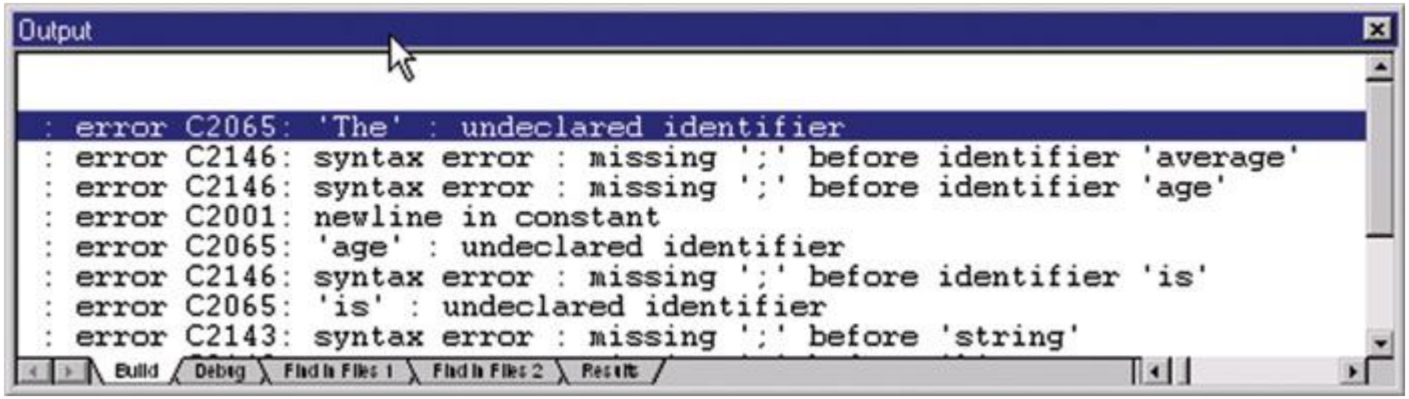
- Phase 1: Needs Analysis
  - Users identify a need
  - Need is clearly defined using tools

Systems Development Life Cycle

- Phase 2: Systems design
  - Solution to the need is defined
  - Many tools are used
  - Prototypes of the solution are built

Systems Development Life Cycle

- Phase 3: Development
  - Solution to the problem is built
  - Programmers write the solution
  - Technical writers document the process
  - Solution is continually tested



```
: error C2065: 'The' : undeclared identifier
: error C2146: syntax error : missing ';' before identifier 'average'
: error C2146: syntax error : missing ';' before identifier 'age'
: error C2001: newline in constant
: error C2065: 'age' : undeclared identifier
: error C2146: syntax error : missing ';' before identifier 'is'
: error C2065: 'is' : undeclared identifier
: error C2143: syntax error : missing ';' before 'string'
```

Systems Development Life Cycle

- Phase 4: Implementation
  - The solution is installed
  - Users are converted to the new system
  - Trainers are important in this phase

Systems Development Life Cycle

- Phase 5: Maintenance
  - IT professionals monitor the product
  - Bugs are found and fixed
  - New features are added

Section 01 Introduction to Programming Language(3)

The Choice of Programming Language 프로그래밍 언어 선택

- Regardless of what language you use, you eventually need to convert your program into machine language so that the computer can understand it. There are two ways to do this:
  - compile the program, interpret the program
- The question of which language is the best is one that consumes a lot of time and energy among computer professionals. Every language has its strengths and weaknesses. For example, FORTRAN is a particularly good language for processing numerical data, but it does not lend itself very well to organizing large programs. Pascal is very good for writing well-structured and readable programs, but it is not as flexible as the C programming language. C++ embodies powerful object-oriented features, but it is complex and difficult to learn.
- The choice of which language to use depends on the type of computer the program is to run on, what sort of program it is, and the expertise of the programmer.

Section 02 Type of Programming Language(1)

C

- C is a high-level programming language developed by Dennis Ritchie at Bell Labs in the mid 1970s. Although originally designed as a systems programming language, C has proved to be a powerful and flexible language that can be used for a variety of applications, from business programs to engineering. C is a particularly popular language for personal computer programmers because it is relatively small - it requires less memory than other languages.
- The first major program written in C was the UNIX operating system. And for many years C was considered to be inextricably linked with UNIX. Now, however, C is an important language independent of UNIX.
- Although it is a high-level language, C is much closer to assembly language than are most other high-level languages. This closeness to the underlying machine language allows C programmers to write very efficient code. The low-level nature of C, however, can make the language difficult to use for some types of applications.

Section 02 Type of Programming Language(2)

C++

- C++ is a high-level programming language developed by Bjarne Stroustrup at Bell Labs. C++ adds object-oriented features to its predecessor, C. C++ is one of the most popular programming language for graphical applications, such as those that run in Windows and Macintosh environments.

Section 02 Type of Programming Language(3)

Java 자바

- Java is a high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.
- Java is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a .java extension) are compiled into a format called byte code (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (JVM), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Bytecode can also be converted directly into machine language instructions by a just-in-time compiler (JIT).

Section 02 Type of Programming Language(4)

C#

- C# is pronounced "see-sharp." A hybrid of C and C++, it is a Microsoft programming language developed to compete with Sun's Java language. C# is an object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications. C# boasts type-safety, garbage collection, simplified type declarations, versioning and scalability support, and other features that make developing solutions faster and easier, especially for COM+ and Web services. Microsoft critics have pointed to the similarities between C# and Java.

Section 03 Object-Oriented Programming (1)

- Object-oriented programming (OOP) is a programming language model organized around “objects” rather than “actions” and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data. The programming challenge was seen as how to write the logic, not how to define the data. Object-oriented programming takes the view that what we really care about are the objects we want to manipulate rather than the logic required to manipulate them. Examples of objects range from human beings (described by name, address, and so forth) to buildings and floors.
- The first step in OOP is to identify all the objects you want to manipulate and how they relate to each other. Once you’ve identified an object, you generalize it as a class of objects and define the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. A real instance of a class is called an “object” or, in some environments, an “instance of a class.” The object or class instance is what you run in the computer. Its methods provide computer instructions and the class object characteristics provide relevant data. You communicate with objects and they communicate with each other - with well-defined interfaces called messages.

Section 03 Object-Oriented Programming (2)

- The concepts and rules used in object-oriented programming provide these important benefits:
  - The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class characteristics. Called inheritance, this property of OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding.
  - Since a class defines only the data it needs to be concerned with, when an instance of that class is run, the code will not be able to accidentally access other program data. This characteristic of data hiding provides greater system security and avoids unintended data corruption.
  - The definition of a class is reusable not only by the program for which it is initially created but also by other object-oriented programs (and, for this reason, can be more easily distributed for use in networks).
  - The concept of data classes allows a programmer to create any new data type that is not already defined in the language itself.
  - Simula was the first object-oriented programming language. C++ and Java are the most popular object-oriented languages today. The Java programming language is designed especially for use in distributed applications on corporate networks and the Internet.

Section 04 Self Check

- Visual Basic is a programming language and environment developed by Microsoft. Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment for developing user interfaces. Instead of worrying about syntax details, the Visual Basic programmer can add a substantial amount of code simply by dragging and dropping controls, such as buttons and dialog boxes, and then defining their appearance and behavior.
- Although not a true object-oriented programming language in the strictest sense, Visual Basic nevertheless has an object-oriented philosophy. It is sometimes called an event-driven language because each object can react to different events such as a mouse click.

Section 05 Practical English(1)

"You're telling me it will take 13 years to install my education! What kind of outdated software is this school using?"

Section 05 Practical English(3)

"Mom says I can only use the computer three hours a day. It takes longer than just to delete my spam!"

응용문장

- I gave her the book.
- I gave her a book.

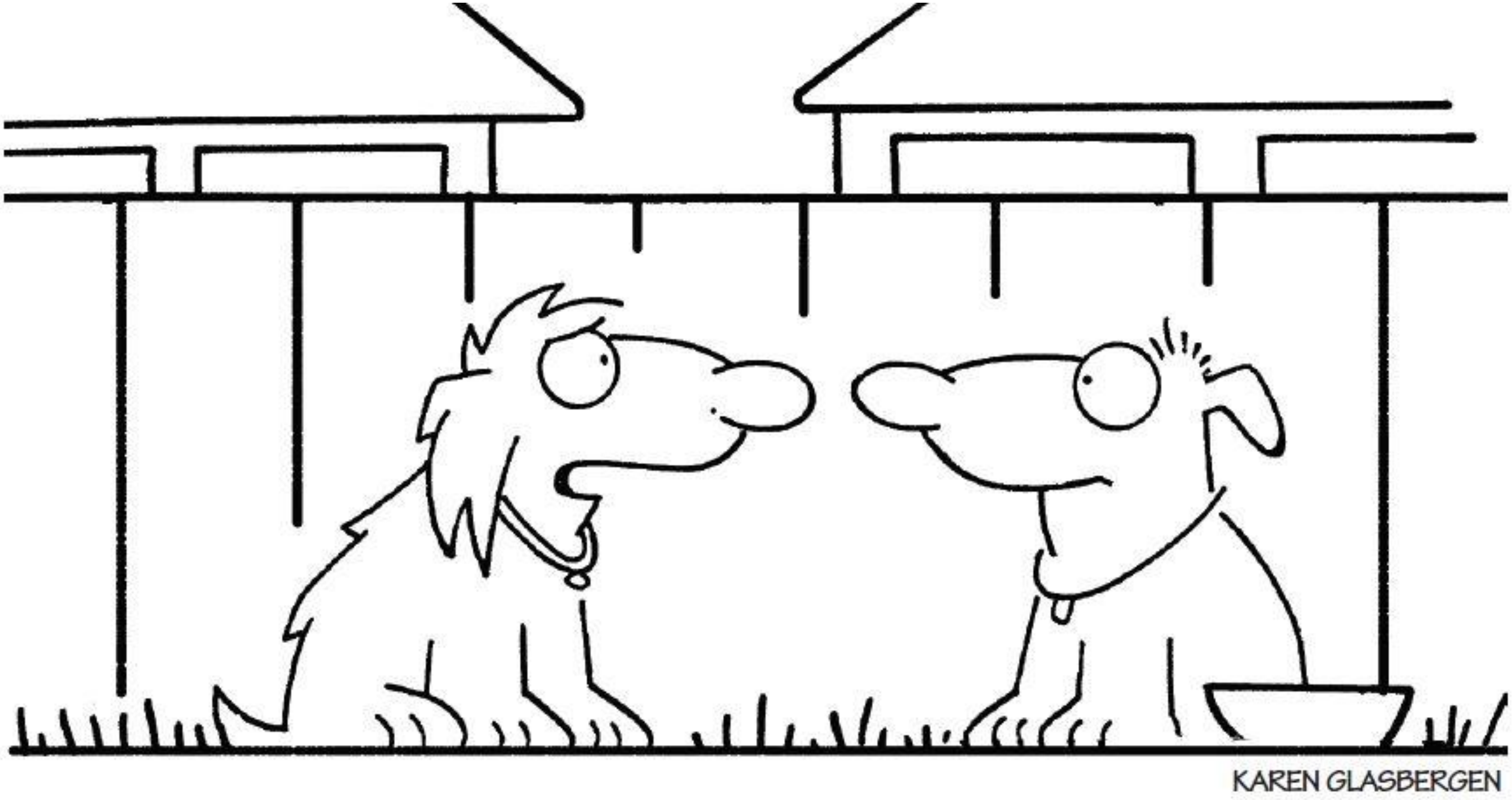
Section 05 Practical English(4)

"No matter how many times I empty the trash, when I go out to the kitchen, the can is still full!"

응용문장

- No matter how hard it is, you have to do your best. I gave her a book.
- No matter what you say, I will go my way.

Section 05 Practical English(5)



"I met someone wonderful in a chat room...
and then I found out she's a cat!"

Section 05 Practical English(6)

MOM
KAREN GLASBERGEN

“My family thinks we have a brand new high resolution monitor – but all I did was wiping off the dust and fingerprints.”

Section 05 Practical English(7)

"Until the virus has been identified and removed, IT has issued an immediate ban on any use of e-mail attachments. For more details, please refer to the attached document."