3 Computer languages

A Read the text. How many high-level computer languages are mentioned?

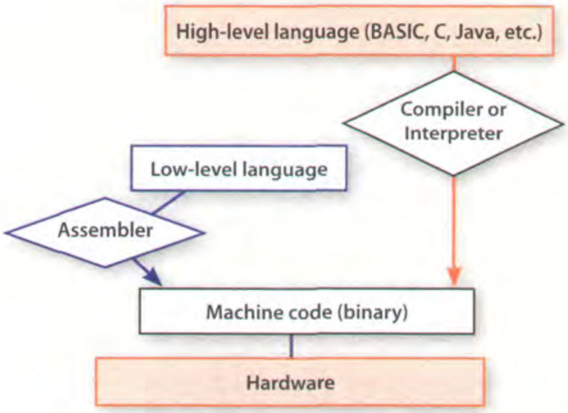
Computer languages

Unfortunately for us, computers can't understand spoken English or any other natural language. The only language they can understand directly is **machine code**, which consists of 1s and 0s (binary code).

Machine code is too difficult to write. For this reason, we use symbolic languages to communicate instructions to the computer. For example, **assembly languages** use abbreviations such as ADD, SUB, MPY to represent instructions. The program is then translated into machine code by a piece of software called an **assembler**. Machine code and assembly languages are called **low-level languages** because they are closer to the hardware. They are quite complex and restricted to particular machines. To make the programs easier to write, and to overcome the problem of intercommunication between different types of computer, software developers designed **high-level languages**, which are closer to the English language. Here are some examples:

- **FORTRAN** was developed by IBM in 1954 and is still used for scientific and engineering applications.
- **COBOL** (**Co**mmon **B**usiness **O**riented **L**anguage) was developed in 1959 and is mainly used for business applications.
- **BASIC** was developed in the 1960s and was widely used in microcomputer programming because it was easy to learn. **Visual BASIC** is a modern version of the old BASIC language, used to build graphical elements such as buttons and windows in Windows programs.
- **PASCAL** was created in 1971. It is used in universities to teach the fundamentals of programming.
- **C** was developed in the 1980s at AT&T. It is used to write system software, graphics and commercial applications. **C++** is a version of C which incorporates object-oriented programming: the programmer concentrates on particular things (a piece of text, a graphic or a table, etc.) and gives each object functions which can be altered without changing the entire program. For example, to add a new graphics format, the programmer needs to rework just the graphics object. This makes programs easier to modify.
- **Java** was designed by Sun in 1995 to run on the Web. Java applets provide animation and interactive features on web pages. (See Unit 25)

Programs written in high-level languages must be translated into machine code by a **compiler** or an **interpreter**. A compiler translates the source code into **object code** – that is, it converts the entire program into machine code in one go. On the other hand, an interpreter translates the source code line by line as the program is running.



It is important not to confuse **programming languages** with **markup languages**, used to create web documents. Markup languages use instructions, known as **markup tags**, to format and link text files. Some examples include:

- **HTML**, which allows us to describe how information will be displayed on web pages.
- **XML**, which stands for **EX**tensible **M**arkup **L**anguage. While HTML uses pre-defined tags, XML enables us to define our own tags; it is not limited by a fixed set of tags.
- **VoiceXML**, which makes Web content accessible via voice and phone. VoiceXML is used to create voice applications that run on the phone, whereas HTML is used to create visual applications (for example, web pages).

```
<xml>

< name> Andrea Finch </name>

< homework> Write a paragraph describing
the C language </homework>

</xml>
```

In this XML example we have created two new tags: <name> and <homework>

B Read the text again and answer these questions.

1 Do computers understand human languages? Why? / Why not?

2 What is the function of an *assembler*?

3 Why did software developers design high-level languages?

4 Which language is used to teach programming techniques?

5 What is the difference between a *compiler* and an *interpreter*?

6 Why are HTML and VoiceXML called *markup* languages?

C Complete these sentences with a computer language from the text.

1 .... allows us to create our own *tags* to describe our data better. We aren't constrained by a pre-defined set of tags the way we are with HTML.

2 IBM developed .... in the 1950s. It was the first high-level language in data processing.

3 .... applets are small programs that run automatically on web pages and let you watch animated characters, play games, etc.

4 .... is the HTML of the voice web. Instead of using a web browser and a keyboard, you interact with a voice browser by listening to pre-recorded audio output and sending audio input through a telephone.

5 This language is widely used in the business community. For example, the statement ADD VAT to NET-PRICE could be used in a .... program.

4 *Word building*

Look at the words in the boxes. Are they nouns, verbs or adjectives? Write *n*, *v* or *adj* next to each word. There may be more than one possible answer. Complete the sentences with words from the boxes.

program .... programmers .... programming .... programmable ....

1 .... is the process of writing a program using a computer language.

2 A computer .... is a set of instructions that tells the computer how to do a specific task.

3 Most computer .... make a plan of the program before they write it.

4 A .... keyboard allows the user to configure the layout and meaning of the keys.

compile .... compiler .... compilation ....

5 Programs written in a high-level language require .... – that is, translation into machine code, the language understood by the processor.

6 A source program is converted into machine code by software called a ....

7 Programmers usually .... their programs to generate an object program and diagnose possible errors.

bug .... debug .... debugger .... debugging ....

8 Any error or malfunction of a computer program is known as a ....

9 A .... is a program used to test and .... other programs.

10 The process of going through the code to identify the cause of errors and fixing them is called ....