

## 1 Programming

**A** In pairs, discuss what you think **programming** is.

**B** Look at the definition of **programming** in the Glossary. Is it similar to yours?

```
#include <stdio.h>
main( )
{
    printf("good morning\n");
}
```

## 2 Steps in programming

**A** Match the words (1–5) with the definitions (a–e).

- 1 flowchart
- 2 source code
- 3 compiler
- 4 machine code
- 5 debugging

- a Program instructions written in a particular computer language
- b The techniques of detecting and correcting errors (or bugs) which may occur in programs
- c A diagram representing the successive logical steps of the program
- d A special program which converts the source program into machine code – the only language understood by the processor
- e The basic instructions understood by computers; it consists of 1s and 0s (binary code)

This C program tells the computer to print the message 'good morning'

**B** Listen to Andrea Finch, a software developer, talking to a group of students on a training course about how a program is written and check your answers to A.

**C** Listen again and put these steps into the correct order.

- Write instructions in a programming language
- Prepare documentation
- Understand the problem and plan a solution
- Make a flowchart of the program
- Compile the program (to turn it into machine code)
- Test and debug the program

**D** Listen again and make detailed notes. In pairs, use your notes to write a short explanation of what each step in C means.

### 3 Computer languages

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

#### Computer languages

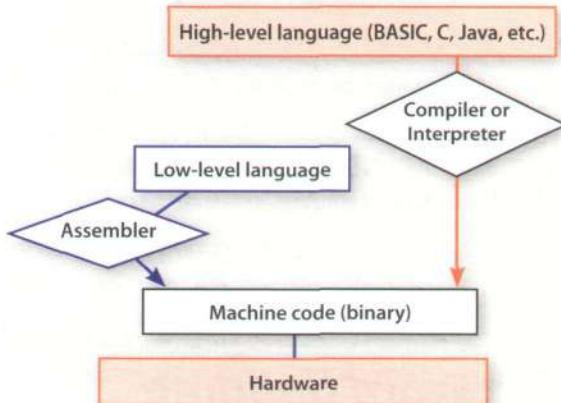
Unfortunately for us, computers can't understand spoken English or any other natural language. The only language they can understand directly is **machine code**, which consists of 1s and 0s (binary code).

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 (Common Business Oriented Language)** 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 EXtensible Markup Language. 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 **XML** ..... 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 **Fortran** ..... in the 1950s. It was the first high-level language in data processing.
- 3 **Java** ..... applets are small programs that run automatically on web pages and let you watch animated characters, play games, etc.
- 4 **VOICE XML** ..... 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 ..... **COBOL** ..... 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 **n**    programmers **n**    programming **v**    programmable **adj**

- 1 **programming** ..... is the process of writing a program using a computer language.
- 2 A computer **program** ..... is a set of instructions that tells the computer how to do a specific task.
- 3 Most computer **programmers** ..... make a plan of the program before they write it.
- 4 A **programmable** ..... keyboard allows the user to configure the layout and meaning of the keys.

compile **v**    compiler **n**    compilation **n**

- 5 Programs written in a high-level language require **compilation** ..... – 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 **compilator** .....
- 7 Programmers usually **compile** ..... their programs to generate an object program and diagnose possible errors.

bug **n**    debug **v**    debugger **n**    debugging **v**

- 8 Any error or malfunction of a computer program is known as a **bug** .....
- 9 A **debugging** ..... is a program used to test and **debug** ..... other programs.
- 10 The process of going through the code to identify the cause of errors and fixing them is called **debugging** .....

## 5 Language work: the infinitive

### A Look at the HELP box and then make sentences using these prompts.

- 1 not easy / write instructions in COBOL  
*It's not easy to write instructions in COBOL.*
- 2 expensive / set up a data-processing area
- 3 advisable / test the programs under different conditions
- 4 unusual / write a program that works correctly the first time it's tested
- 5 important / use a good debugger to fix errors
- 6 easy / learn Visual BASIC

### B Choose the correct words (a–c) to complete these sentences.

- 1 We use high-level languages because machine code is too difficult \_\_\_\_\_, understand and debug.  
a read    b reading **c** to read
- 2 I went on the course \_\_\_\_\_ how to be a better programmer.  
a learn **b** to learn    c for to learn
- 3 I'm not interested in \_\_\_\_\_ that computer language.  
a learn **b** learning    c to learn
- 4 He refuses \_\_\_\_\_ the project with me.  
a do    b doing **c** to do
- 5 The engineers warned the employees not \_\_\_\_\_ the cables.  
a touch    b touching **c** to touch
- 6 They may not \_\_\_\_\_ to the conference.  
**a** come    b coming    c to come
- 7 Spyware can make your PC \_\_\_\_\_ more slowly.  
**a** perform    b performing    c to perform
- 8 This program is too slow \_\_\_\_\_ the simulation.  
a do **b** to do    **c** for doing

### HELP box

#### The infinitive

The infinitive with *to* is used in the following ways:

- To express purpose

*We use symbolic languages **to communicate** instructions to the computer.  
(=in order to communicate ...)*

**Not:** ... **for** to communicate

- After adjectives

*BASIC was widely used in the past because it was **easy to learn**.*

*Machine code is too **difficult to write**.  
(=not easy enough to write)*

- After certain verbs (e.g. **afford, demand, plan, agree, expect, promise, appear, hope, refuse, arrange, learn, try, decide, manage**)

*A lot of companies are now **trying to develop** voice applications for web access.*

- After the object of certain verbs (e.g. **advise, encourage, allow, expect, tell, ask, invite, want, enable, order, warn**)

*HTML **allows us to describe** how information will be displayed on web pages.*

The bare infinitive (without *to*) is used in the following ways:

- After modal verbs (e.g. **can, could, may, might, will, would, must, should**)

*Unfortunately, computers **can't understand** spoken English.*

*High-level languages **must be** translated into machine code.*

- After the object with the verbs **make** and **let**

*Programs **make computers perform** specific tasks.*



### C In pairs, discuss something

- 1 you can't afford to buy at the moment.
- 2 you've arranged to do this weekend.
- 3 you've learnt to do in the last year.
- 4 you'd advise someone to do before buying a new PC.
- 5 you'd expect to be included with an anti-virus package.
- 6 you can do with Java applets.

## 6 Visual BASIC and VoiceXML

A Work in pairs. Student A reads about Visual BASIC, Student B reads about VoiceXML. Try not to look at your partner's text. Complete your part of the table.

#### Student A

**Visual BASIC** was developed by Microsoft in 1990. The name **BASIC** stands for Beginner's All-purpose Symbolic Instruction Code. The adjective **Visual** refers to the technique used to create a graphical user interface. Instead of writing a lot of instructions to describe interface elements, you just add pre-defined objects such as buttons, icons and dialog boxes. It enables programmers to create a variety of Windows applications.

#### Student B

**VoiceXML** (EXtensible Markup Language) was created in 2000 to make web content accessible via the telephone. For input, it uses voice recognition. For output, it uses pre-recorded audio content and text-to-speech. Applications:

- voice portals, where you can hear information about sports, news, traffic, etc.
- voice-enabled intranets (private networks)
- voice e-commerce
- home appliances controlled by voice

	Visual BASIC	VoiceXML
What does Visual BASIC / VoiceXML stand for?		
When was it developed?		
What are its main features?		
What is it used for?		



B Ask your partner about the other language and complete the table.

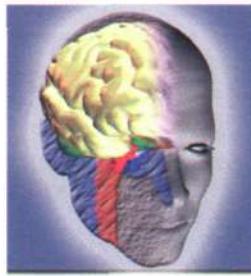
## 1 Java applets

**A Match the examples of Java programs, known as *applets*, (a–e) with the descriptions (1–5).**

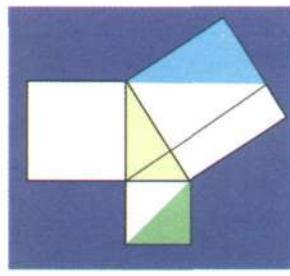
- 1 This Land Rover applet allows you to change the look of the vehicle.
- 2 The Pythagoras theorem applet gives the proof of the Pythagorean theorem without words. It allows you to manipulate triangles and go through the steps of the geometrical proof.
- 3 The Jman for Java applet permits medical researchers to view sequential MRI (Magnetic Resonance Images) of the brain.
- 4 An analogue clock applet displays the time according to the web user's computer and lets you set the colours and style of the hands and numbers.
- 5 A banner applet displays graphic images on websites in order to advertise products or services.



a \_\_\_\_\_



c \_\_\_\_\_



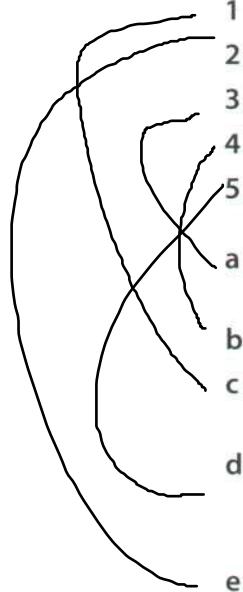
e \_\_\_\_\_



b \_\_\_\_\_



d \_\_\_\_\_



**B Match the terms (1–5) with the definitions (a–e).**

- 1 Java
- 2 applet
- 3 plug-in
- 4 platform-independent
- 5 object-oriented programming

- a an auxiliary program that enables web browsers to support new content, for example animation
- b software that can run on any operating system
- c an island in Indonesia, coffee (in American slang), and a programming language for internet applications
- d a computer programming technique that allows the creation of objects that interact with each other and can be used as the foundation of others; used to create graphical user interfaces
- e a small Java application, usually designed to run automatically within a web page



The Java logo

## 2 The Java language

A These statements about Java are all false. Read the text and correct them.

- 1 Java was invented by Microsoft.
- 2 With the interpreter, a program is first converted into Java bytecodes.
- 3 Java is not compatible with most computing platforms.
- 4 The Java language is single-threaded, one part executing at a time.
- 5 Java has no competitors.
- 6 Flash files are called *animations*.

### The Java language

Java is a programming language developed by Sun Microsystems, specially designed to run on the Web. Java programs (called **applets**) let you watch animated characters and moving text, play music, and interact with information on the screen (for example, control animations and select options).

#### Characteristics of the Java language

Java is an **object-oriented** language, similar to C++, but more dynamic and simplified to eliminate possible programming errors. A Java program is both compiled and interpreted (see Unit 24). First, the source code (a file with a **.java** extension) is compiled and converted into a format called bytecode (a file with a **.class** extension), which can then be executed by a Java interpreter (see Fig. 1). Compiled Java code can run on most computers because there are Java interpreters, known as **Java Virtual Machines**, for most operating systems.

Java is **multi-threaded**, meaning a Java program can have multiple threads (parts) – that is, many different things processing independently and continuously. This enables the program to make the best use of available CPU power.

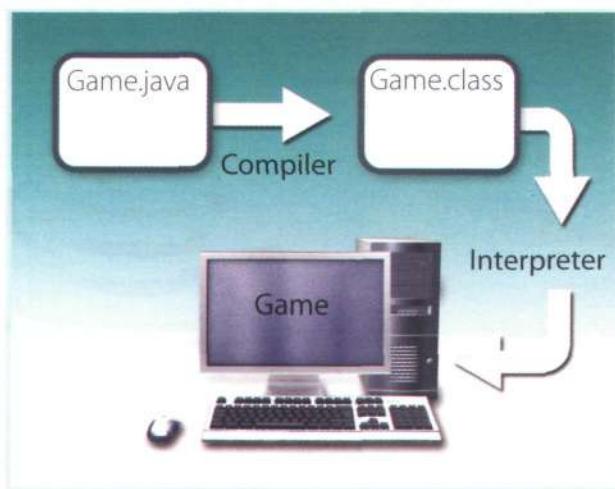


Fig. 1

#### Why is Java popular?

Most programmers like Java because it allows them to write applets which make web pages more interactive and attractive. They can create graphical objects (for example, bar charts and diagrams) and new controls (for example, check boxes and push buttons with special properties). A web page that uses Java can have sounds that play in real time, music that plays in the background, cartoon-style animations, real-time video and interactive games.

The Java Micro Edition platform (**Java ME**) is used in mobile devices. It provides flexible tools to create applications that run on mobile phones, PDAs, TV set-top boxes and printers. Nowadays, most phones are configured to use Java games.

#### Alternatives to Java

One alternative to Java is Microsoft's **C#**, pronounced 'C sharp', a **.NET** language based on C++ with elements from Visual Basic and Java. There are no substantial differences between C# and Java. When software developers do measurements on pieces of code, sometimes Java is faster, sometimes C# is.

Another competitor is Adobe **Flash** technology, which supports graphics, a scripting language called ActionScript, and the streaming of audio and video. Flash is used to create animation and advertisements, to integrate video into web pages, and to develop rich internet applications such as portals. **Flash files**, traditionally called **flash movies**, have a **.swf** file extension. They may be an object on a web page or be played in the stand-alone Flash Player.

### B Match the words (1–6) with the words (a–f) to make technical terms from the text.

- |               |   |             |   |          |            |
|---------------|---|-------------|---|----------|------------|
| 1 Java        | a | 4 web       | b | a applet | d system   |
| 2 operating   | d | 5 source    | c | b page   | e object   |
| 3 programming | f | 6 graphical | e | c code   | f language |

### C Complete the sentences with words from the box.

interpreted animated configured used pronounced object-oriented compiled

- 1 Java lets you watch animated characters on web pages.
- 2 Java is an object-oriented language, similar to C++ but more dynamic.
- 3 First, the source code of a Java program is compiled into an intermediate format called *bytecode*. This is then used by any system possessing a Java interpreter.
- 4 The Java ME platform is widely used in mobile devices.
- 5 Nowadays, most mobile phones are interpreted to use Java games.
- 6 Microsoft's C# is a simplified version of C and C++ for the Web. It's pronounced 'C sharp'.

## 3 Language work: the -ed form

### A Look at the HELP box and then put these verbs into the correct column.

stopped	asked
described	decided
produced	called
watched	executed
published	object-oriented
programmed	persuaded
configured	converted
arranged	designed

/t/	/d/	/ɪd/

### HELP box

#### The -ed form

We use the **-ed** form in the following ways:

- To make the past simple (affirmative) of regular verbs

*Sun Microsystems developed Java in 1995.*

Remember that not all verbs in the past simple end in **-ed**. See page 166 for a list of irregular verbs. See Unit 19 for more about the past simple.

- To make the past participle of regular verbs

*Flash is used to create animation.*

- To make the adjectival form of some verbs

*Java applets let you watch animated characters.*

The **-ed** is pronounced as:

- /t/ after voiceless sounds: /p/, /k/, /θ/, /s/, /f/, /ʃ/ or /tʃ/. (e.g. *developed, talked, pronounced*)
- /d/ after voiced sounds: /b/, /g/, /ð/, /z/, /v/, /dʒ/, /l/, /r/; nasal consonants: /m/, /n/, /ŋ/; and vowels (e.g. *compiled, designed, simplified*)
- /ɪd/ after /t/ or /d/ (e.g. *interpreted, multi-threaded*)

**B** Complete this extract from a lecture handout about Java with the correct form of the verbs in the box.

call    be    begin    can    decide    rename    have    support    develop    base

The idea for Java started in 1990, when a team of software engineers at Sun Microsystems (1) decided to create a language for a handheld device that could control and interact with various kinds of electronic appliances, ranging from Nintendo Game Boys to VCRs and TV set-top boxes. They (2) developed an object-oriented programming language that one of the engineers, James Gosling, (3) called *Oak*, after the tree outside his window. The device even (4) had an animated character named *Duke*, who would go on to become Java's mascot.

With the advent of the Web in 1993, the company made a web browser (5) supported on the Oak language. Later on, this language was adapted to the Internet and (6) renamed Java. The 1.0 version of Java was officially introduced by Sun in May 1995.

At that time, web pages (7) could only display text, pictures and hyperlinks. With the arrival of Java, web designers (8) were able to include animation and interactive programs on web pages. The first major application created with Java was the HotJava browser. The Java language (9) began to attract serious attention from the internet community and was soon (10) based by Netscape Navigator and MS Internet Explorer. Today, Java is a hot technology that runs on multiple platforms, including smart cards, embedded devices, mobile phones and computers.

**C**  Listen to an extract from the lecture and check your answers to C. Listen carefully to the pronunciation of the verbs that end in -ed.

## 4 Your experience with computers

**A** Make notes about the different stages in your computer history.  
Add more stages if you want to.

Example: 1990: Played my first computer game. It was ...

Possible stages:

- First computer game
- First computer lesson at school/college
- First programming language learnt
- First software used
- First computer course/qualification
- First job involving computers
- First steps on the Internet
- First chat online

### Useful language

When did you first ...?

How long ago did you ...?

How old were you when ...?

I started ... in...

I learnt ... when I was ...

I didn't use the Internet until ...

**B**  Ask a partner about their computer history.  
Look at the *Useful language* box to help you.

