UNIT 11

Program design and computer languages

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)

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

3 Write instructions in a programming language

6 Prepare documentation

1 Understand the problem and plan a solution

2 Make a flowchart of the program

4 Compile the program (to turn it into machine code)

5 Test and debug the program

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.

Computer languages

■**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. 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).

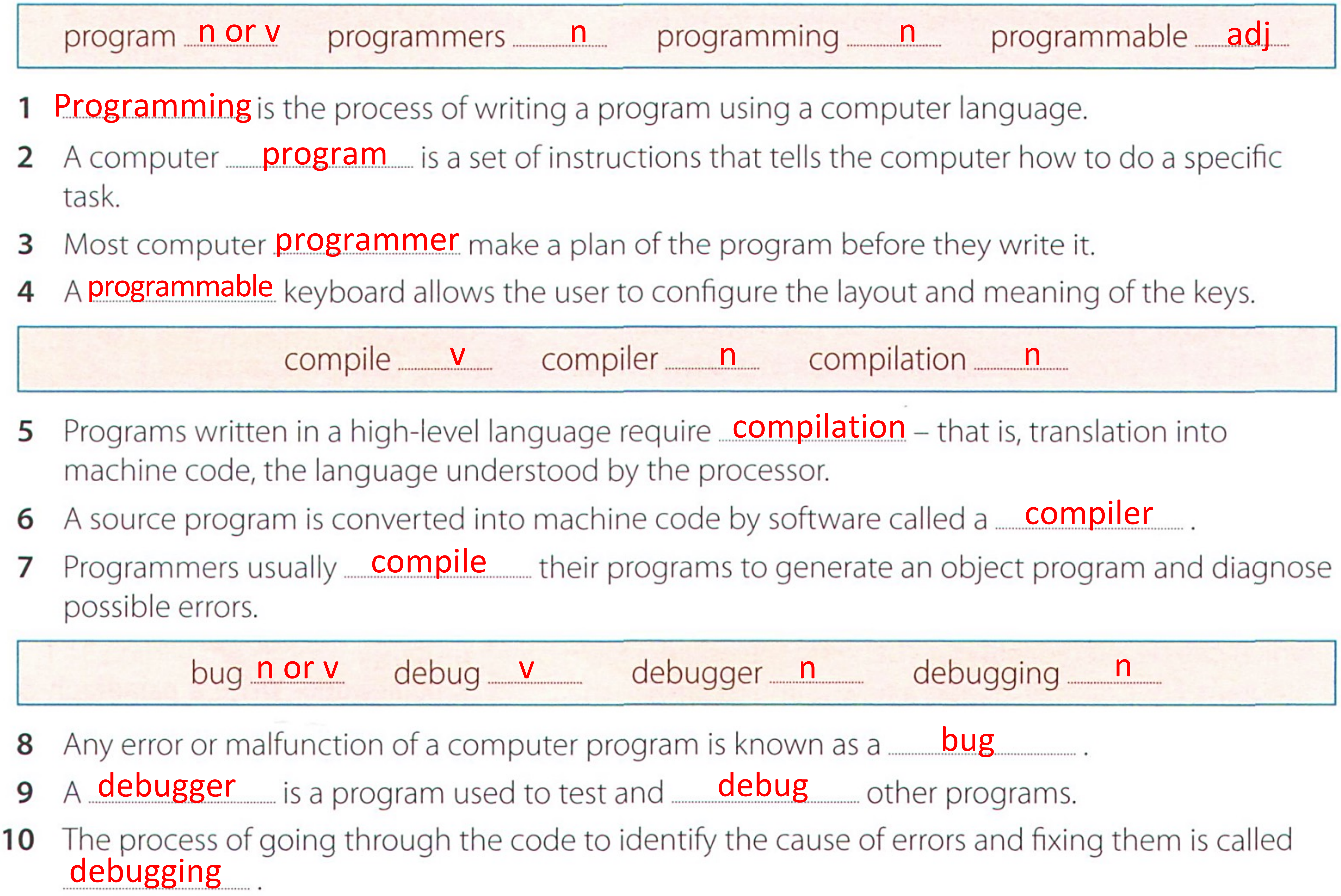
B Read the text again and answer these questions.

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.

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-levellanguage 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. _VoiceXML___ 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-recordedaudio 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.

Are the words in the boxes nouns, verbs or adjectives? Write n, v or adj next to each word. Complete the sentences with words from the boxes.



program n or v programmers n programming n 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 programmer 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 compiler .

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

bug n or v debug v debugger n debugging n

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

9 A debugger 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 .

Động từ nguyên thể

- Để thể hiện mục đích:

  *We use symbolic languages **to communicate** instructions to the computer.*

  (= in order to communicate...)

- Sau tính từ

  *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)*

- Sau một số động từ (ví dụ **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.*

- Sau đối tượng của một số động từ nhất định (ví dụ **advise, encourage, allow, expect, tell, ask, invite, want, enable, order, warn**)

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

- Sau động từ khuyết thiếu (**can, could, may, might, will, would, must, should**)

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

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

- Sau đối tượng của các động từ **make, let**

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

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

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.

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

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	*Student B*
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.	**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?		

- **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.