**Unit N°1 - Programming design and computer languages**

**SPEAKING.** Group work. In groups of four discuss what you think programming is and write a definition.

**VOCABULARY.** Steps in programming. Match the words (1-5) to their definitions (a-e)

1. flowchart
2. sourcecode
3. compiler
4. machine code
5. debugging
6. Program instructions written in a particular computer language.
7. The techniques of detecting and correcting errors (or bugs) which may occur in programs.
8. A diagram representing the successive logical steps in a program.
9. A special program which converts the source program into machine code - the only language understood by the processor.
10. The basic instructions understood by computers; it consists of 1s and 0s (binary code).

**LISTENING.**

a) Read and put these steps into the correct order according to your knowledge. Then, 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. (Infotech - track 29)

\_\_\_ Write instructions in a programming language.

\_\_\_ Prepare documentation.

\_\_\_ Understand the problem and plan the solution.

\_\_\_ Make a flowchart of the program.

\_\_\_ Compile the program.

\_\_\_ Test and debug the program.

b) Listen again and make detailed notes. Then share them with the whole class.

**READING COMPREHENSION**

a) Read the text. How many high-level 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 Os (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.  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). |

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.

**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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**GRAMMAR.** the infinitive

***Reflect upon grammar.***Look back at the text “Computer languages” and underline all the “to + verb” that you can find. Then choose the correct option(s) to complete the sentence.

* We use “to” + an infinitive verb…

1. to express the purpose of something.
2. after adjectives.
3. as the main verb of a sentence.

|  |
| --- |
| **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*.  *We use symbolic languages* ***in order to*** *communicate instructions to the computer*.  *We use symbolic languages* ***~~for to~~*** *communicate instructions to the computer*.   * 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.* |

**LANGUAGE WORK:** the infinitive

After dealing with the theory, 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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Graphics and design**

**SPEAKING**. Discuss in pairs. As a programmer, is it important to know about graphic programmes? If not, who would be in charge of the graphics?

**LISTENING.**

a) Listen to an extract from an online tutorial about graphics programs and answer these questions. Remember to focus on key words and write them down to answer the questions. (Infotech. Track 25)

1 What is a toolbox in graphics software?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

2 What are graphics primitives?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

3 What sort of attributes, or characteristics, can be used in graphical objects?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

4 What does translation mean?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

b) Listen again and complete this extract from the web version of the tutorial.

Graphics programs usually have a toolbox - a collection of drawing and (1) \_\_\_\_\_\_\_\_\_ tools that enable you to type, (2) \_\_\_\_\_\_\_\_\_\_ draw, paint, edit, move, and view images on the computer.

The basic shapes which are used to (3) \_\_\_\_\_\_\_\_\_\_ graphical objects are called primitives. These are usually geometric, such as lines between two points, arcs, circles, polygons, ellipses and even text. Furthermore, you can specify the attributes of each primitive, such as its colour, line type, fill area, interior style and so on.

The various tools in a toolbox usually appear together as pop-up icons in a menu or palette. To use one, you activate it by (4) \_\_\_\_\_\_\_\_ on it. For example, if you want to (5) \_\_\_\_\_\_\_\_ a rectangle, you activate the rectangle tool, and the pop-up options give you the possibility of (6) \_\_\_\_\_\_\_\_\_ rectangles with square or rounded corners.

You can transform an object by translating, (7) \_\_\_\_\_\_\_\_\_ or scaling it. Translation means moving an object to a different location. Rotation is (8) \_\_\_\_\_\_\_\_\_\_\_ the object around an axis. For example, you may need to rotate an object 90 or 180 degrees to fit the drawing. (9) \_\_\_\_\_\_\_\_\_ is making the object larger or smaller.

**GRAMMAR.** The -ing form

***Reflect upon grammar.*** Read the text above (ex b) again and underline all the words that contain an *-ing suffix*. Are they used as verbs, adjectives or nouns? Provide examples for your answers.

|  |
| --- |
| **The -ing form**  We use the -ing form in three ways:   1. **Rendering** includes **lighting** and **shading**.   **Rendering** is a gerund acting as the subject of the sentence. **Lighting** and **shading** are also gerunds, but acting as the object of the sentence.   * A***gerund*** is a word that is formed by a verb + -ing suffix used in a place where we traditionally would find a noun. * They can be the subject or object of a sentence or they can be found after a preposition.   Designers start a project **by making** a wireframe.   * Some verbs are always followed by gerunds (e.g. **avoid, fancy, finish, give up, hate, imagine, involve, keep, look forward to, mind, suggest, enjoy, love,** etc)  1. They use special applets to create **amazing** fractals.   Amazing is an adjective.   1. We are **designing** a new graphic on this computer.   **Designing** is a ***present participle***. Present participles are used in continuous tenses and reduced relative clauses. |

**LANGUAGE WORK.** the -ing form.

a) After dealing with the theory, decide if the -ing forms in these sentences are gerunds, present participles or adjectives. Write **g**, **pp** or **a**.

**1** PCs generate graphics by performing mathematical calculations on data. \_\_\_\_

**2** Businesspeople use graphics to make information more interesting visually. \_\_\_\_

**3** Graphs and diagrams can be more effective ways of communicating with clients than lists of figures. \_\_\_\_

**4** She is designing a logo for the company. \_\_\_\_

**5** If you need to make a presentation, I suggest using PowerPoint. \_\_\_\_

**6** The Internet is a network linking other networks. \_\_\_\_

b) Correct the mistakes in these sentences. There are seven mistakes in total.

**1** Computer animation is the process of create objects which move across the screen.

**2** Texturing involves add paint, colour and filters to drawings and designs.

**3** You can open the colour palette by click on the corresponding icon.

**4** CAD programs are very fast at to perform drawing functions.

**5** A lot of time and money is saved by test a car design before to make the product.

**6** To render refers to the techniques used to make realistic images.

c) The infinitive or the -ing form? 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** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the IT industry is usually well paid.

**a** work **b** working **c** to work

**8** Spyware can make your PC \_\_\_\_\_\_\_\_\_\_\_\_\_ more slowly.

**a** perform **b** performing **c** to perform

**9** This program is too slow \_\_\_\_\_\_\_\_\_\_\_\_\_ the simulation.

**a** do **b** to do **c** for doing

**10** Some people are desperate about \_\_\_\_\_\_\_\_\_\_\_ money more than learning how to program.

**a** making **b** to make **c** make

**SPEAKING.** Work in pairs. Student A reads about Python and [Student B](#_d0225x1r4bpr) reads about Javascript. Do not read your partner’s text. Then ask and answer questions to complete the chart.

|  |
| --- |
| **Student A**  Python was created by Guido van Rossum and first released in 1991. The name Python is inspired by the British comedy group Monty Python. Python is known for its clear syntax and readability, often described as "executable pseudocode." While not inherently "visual" in the same way as Visual Basic, Python offers powerful libraries like Tkinter, PyQt, and Kivy for creating graphical user interfaces. Instead of a drag-and-drop interface, developers use code to define and arrange interface elements such as buttons, labels, and windows. Python's versatility enables programmers to build a wide range of applications, from web applications and data analysis tools to desktop applications and games. |

|  |  |  |
| --- | --- | --- |
|  | **Student A** | **Student B** |
| Who developed the programming language and in what year was it first released? |  |  |
| What does the name of the language (or a part of it) stand for or what inspired it? |  |  |
| How is the language used to create user interfaces, and what are some examples of interface elements that can be created? |  |  |
| What types of applications can programmers create using this language? |  |  |

**Working in the IT industry.**

**READING.** Read the memo. What is Lila writing about?

|  |
| --- |
| FROM: Chief Information Officer  TO: All Staff  DATE: June, 17th  SUBJECT: XBM Digital rules  Dear all,  As you know, many new staff joined us at XBM Digital, so now is a good time for a reminder about some of our rules.  Firstly, passwords are important to keep our network secure. Natasha is the person who sets the passwords and usernames at first. As soon as she gives them to you, you **need to** change the passwords which **have to** be alphanumeric and at least eight characters long. Also, you **should not** share them with anyone. You **might** find it difficult, but you **can** ask Natasha for help.  Secondly, the hardware that we have in our offices is delicate. You **mustn't** eat or put drinks near the computers. You **can** take short breaks for eating and drinking. You **must** do it in the dining room where all our staff eat everyday . You **don’t have to** spend time cleaning your stuff, Rudy and Mike **will** take care of that for you.  Best regards,  Lila Davis |

**GRAMMAR**. Modal verbs

FORM. Look at the words in **bold** and complete the rule.

* The verb that follows a modal verb is...

1. Present participle
2. Past participle
3. Infinitive verb

FUNCTION. Complete the chart with examples from the text and analyse the function of each modal verb.

|  |  |
| --- | --- |
| ***Example*** | ***Function*** |
| you **(1) \_\_\_\_\_\_ \_\_\_\_\_\_\_** it in the dining room…  you **(2) \_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_**  the passwords | Personal obligation / strong necessity |
| …which **(3) \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_** alphanumeric… | External obligation |
| You **(4) \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**  or put drinks near the computers. | Prohibition |
| You **(5) \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_** time cleaning your stuff | Lack of necessity / Lack of obligation |
| you **(6) \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_** them with anyone.  You should buy a new computer. | Advice |
| You **(7) \_\_\_\_\_\_\_\_ \_\_\_\_\_\_**short breaks for eating and drinking.  I **could not** swim when I was a child. (Past)  You **will be able to** program in nearly every language when you finish this course. (future) | Ability / inability |
| You **(8) \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_** it difficult  I **may** go to the party, but I still don’t know. | Possibility |
| **Can** I go to the toilet, please?  **Could** you tell me where the restroom is?  **May** I help you? | Request / Asking for permission / ask for or offer help |

LANGUAGE WORK.

a) Complete the sentences with the appropriate modal verbs. There’s an extra modal verb that you don’t need to use:

**Should - can - don’t have to - might - have to - must - mustn’t - doesn’t have to**

1. We \_\_\_\_\_\_\_\_\_\_ have a picnic tomorrow, but it depends on the weather. (possibility)
2. You \_\_\_\_\_\_\_\_\_\_\_ wear a suit to the interview. (Advice)
3. Mary! you \_\_\_\_\_\_\_\_\_\_\_\_\_\_ come to your aunt’s birthday next Saturday night. (Lack of necessity)
4. You \_\_\_\_\_\_\_\_\_\_ smoke here! (Prohibition)
5. \_\_\_\_\_\_\_\_\_\_\_ I go to the toilet, please? (request/ask for permission)
6. Lisa \_\_\_\_\_\_\_\_\_\_ tidy her room before she goes out! (personal obligation)
7. In Britain you \_\_\_\_\_\_\_\_\_\_ drive on the left. (external obligation)

b) Write sentences...

a) ...giving **advice** to a friend as regards the clothes to wear on a party.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) ...**asking for permission** to dosomething in a **polite** way.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c) ...describing a certain **ability** of your friend.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d) ...a rule that you obey at work/university. (**external obligation**)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

e) ...something that you do because of **personal obligation**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

f) ...something that is **prohibited** to you.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

g) ...something that you don't need to do. (**lack of necessity**)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**GRAMMAR.** Relative clauses

What is the difference between a sentence and a clause? Now, look at the underlined words in the memo above and choose the correct options to complete the rules.

1. The words **that, where, who** and **which**…

1. Introduce a new clause with information about something mentioned previously.
2. Introduce a new clause with information about something we didn’t mention.

2. The relative clauses are found…

1. At the beginning of a sentence.
2. Following a previous clause or noun phrase.

3. The relative clause can be…

1. part of the subject.
2. part of the predicate.
3. both previous options.

4. We use the relative pronoun **who** (that can usually be replaced by **that)**…

1. To refer to things.
2. To refer to people.

5. We use the relative pronoun **which** (that can usually be replaced by **that)**…

1. To refer to things.
2. To refer to people.

6. We use the relative pronoun **where** (that **can’t** be replaced by that)...

1. as the subject of the relative clause
2. followed by a subject in the relative clause.

|  |
| --- |
| **Defining and non-defining relative clauses**   * We use **defining relative clauses** to give essential information about someone or something. It usually comes immediately after the noun it describes. We don’t use commas to mark its beginning or end.   *She is the woman* ***who*** *I met with yesterday.*   * We often leave out the relative pronoun when it is the object of the verb:   *They’re the people (that/who) she met at Jon’s office.*  *Here are some bugs (that/which) the developer has identified.*   * We use **non-defining relative clauses**to give extra information about the person or thing. It is not necessary information. We don’t need it to understand who or what is being referred to. We use commas to mark the beginning and end of the non-defining relative clause. We can’t use the relative pronoun ‘that’ to introduce a non-defining relative clause.   *Clare****,*** *who I work with****,*** *is in charge of the project.* |

**LANGUAGE WORK**. Relative clauses

a) Complete the sentences with *who, which* or *where.*

1. Do you know the man \_\_\_\_\_\_\_\_ lives next door?
2. That’s the gallery \_\_\_\_\_\_\_\_\_\_ had the Leonardo Da Vinci exhibition.
3. Are those the people \_\_\_\_\_\_\_ are selling their house?
4. Do you know a good restaurant \_\_\_\_\_\_\_\_\_ is open on Saturday night?
5. Is that the bus \_\_\_\_\_\_\_\_\_\_ goes to the airport?
6. We walked past the school \_\_\_\_\_\_\_\_\_\_\_ their children go.
7. She’s the woman \_\_\_\_\_\_\_ everyone is talking about.
8. I took my laptop back to the shop \_\_\_\_\_\_\_\_\_\_\_ I bought it.
9. Antwerp is the city \_\_\_\_\_\_\_\_\_ I lived as a child.
10. Is there someone \_\_\_\_\_\_\_\_\_ can speak Arabic in your class?

b) Write sentences with ***defining relative clauses*** using the prompts given:

1. Bus = central cable / link / computers

\_\_\_ A bus is a central cable which/that links computers \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Customers = people / buy / things

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Production facilities = places / manufacture / products

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. WAN = type of network / cover / large areas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**WRITING.** Describing functions and features.

Read the instructions below and write a description of the Sony PlayStation 3 controller.

|  |
| --- |
| **Describing functions and features**  We can describe functions and features by using:   * a preposition + a gerund   This is a device **for controlling** the cursor and selecting items on the screen.   * An infinitive to express purpose   It is used **to control**…   * A relative clause   This device**, which is an input device,** is used to…  This is a device **which** is used to…   * A modal verb + bare infinitive   You **can connect** it to the computer…  We can also describe features like this:   * An optical mouse **has** an optical sensor instead of a ball underneath. * It usually **features** two buttons and a wheel. * A wireless mouse **operates/works** without cables. * It **allows** the user **to** answer multiple-choice questions… |