Co-funded by the Erasmus+ Programme of the European Union

2020-1-UK01-KA227-YOU-094543

IO1- A3: Digital Design Compendium
Module: Computer programming

KA2 - Cooperation for innovation and the exchange of good practices
Partnerships for creativity

REVISION HISTORY

Version	Date	Author	Description	Action	Pages
1.0	15/10/2021	CIVIC	Creation	C	12

(*) Action: C = Creation, I = Insert, U = Update, R = Replace, D = Delete

REFERENCED DOCUMENTS

ID	Reference		Title
1	2020-1-UK01-KA227-YOU-094543		HerTour4Youth Proposal
2			

APPLICABLE DOCUMENTS

ID	Reference		Title
1	Deliverable IO1.A2		Elaboration of Training Methodology
2			

Contents

1. Introduction 4
1.1 Learning Outcomes 4
1.2 Key words 4
1.3 Estimated seat time 4
1.4 Glossary of terms 4
2. Computer programming 7
2.1 Introduction 7
2.2 Computer programmer 9
2.1.1 Programming languages 11
2.1.2 Programming process 13
2.1.3 Programming principles 17
2.1.4 How can you become a computer programmer? 18
3. Assessment 21
3.1 Knowledge assessment 21
3.2 Skills assessment 23
4. References 24

1. Introduction

If you're wondering what computer programming is, just imagine a very simple computer machine in your house (it can be a thermostat, your connection router etc.). Now, just know that that computer doesn't work on its own, but it has been programmed by someone. That someone is a computer programmer. Every computer needs a set of instructions to function smoothly. That is what computer programming is all about: it can consist of a simple set of instructions to facilitate specific actions to a complex set of instructions involving reading and sorting out data.

1.1 Learning Outcomes

After completing this module, you will be able to:

-Have a basic understanding of programming principles
-Have a deeper knowledge of how ICT tools work
-Be able to apply programming principles

1.2 Key words

- **ICT**
- **Computer programmer**
- **Programming languages**

1.3 Estimated seat time

1 hour.

1.4 Glossary of terms

ICT: Stands for "Information and Communication Technologies." ICT refers to technologies that provide access to information through telecommunications. (TechTerms, 2021)

Computer programmer: a computer programmer is the person responsible to create instructions for a computer to execute by writing and testing code that enables applications and software programs to operate successfully. (Techopedia, 2014)

Programming languages: computer programming languages are any of the various languages for expressing a set of detailed instructions for a digital computer. (D. Hemmendinger, 2021)

Analog: Adjective that refers to a mechanism or device in which information is represented by continuously variable physical quantities. It's opposed to digital. (Merriam Webster, 2021)

Artificial Intelligence: Artificial intelligence leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind, without confining itself to methods that are biologically observable." (IBM, 2020)

Machine Learning: Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. (IBM, 2020)

Binary: Binary is a base-2 number system invented by Gottfried Leibniz that's made up of only two numbers or digits: 0 (zero) and 1 (one). This numbering system is the basis for all binary code, which is used to write digital data such as the computer processor instructions used every day. (ComputerHope, 2021)

Co-funded by the
Erasmus+ Programme
of the European Union

HerTour4Youth

2020-1-UK01-KA227-YOU-094543

2. Computer programming

2.1 Introduction

In our computer-led world, we know computers can do lots of remarkable things. The rapid technological advancement we've witnessed over the last few decades has a wide-reaching effect on how everything works. You have to look below the surface to see where the most significant changes took place. Computers replaced millions of working hours and warehouses of analog machines with faster, safer, and more reliable systems.
Since computers run on code, it's apparent why you can find it anywhere. Computers will continue to replace outdated technology in everything from microwaves to power plants. And the presence of code in our daily lives will increase.

Also almost everyone nowadays owns a laptop that that can be used for simple computer processes, such as creating a spreadsheet or creating simple documents. However, there are also people who use their laptop in a more complex and professional way. That might be required for completing millions of financial transactions a day and controlling the infrastructure that makes modern life possible.
With that said, it is important to remember that no computer can do anything if it's not programmed by a human being. In this case we're talking about the figure of the computer programmer.

The computer programmer is the professional figure responsible for writing codes, or *coding*, that instruct how a computer, application or software program should perform.

As mentioned before this set of instruction can be quite simple and the whole coding process might just require adding two numbers, but it also could involve a more complex as reading data from temperature sensors to adjust a thermostat, sorting data to complete intricate scheduling or critical reports or taking players through multi-layered worlds and challenges in games.

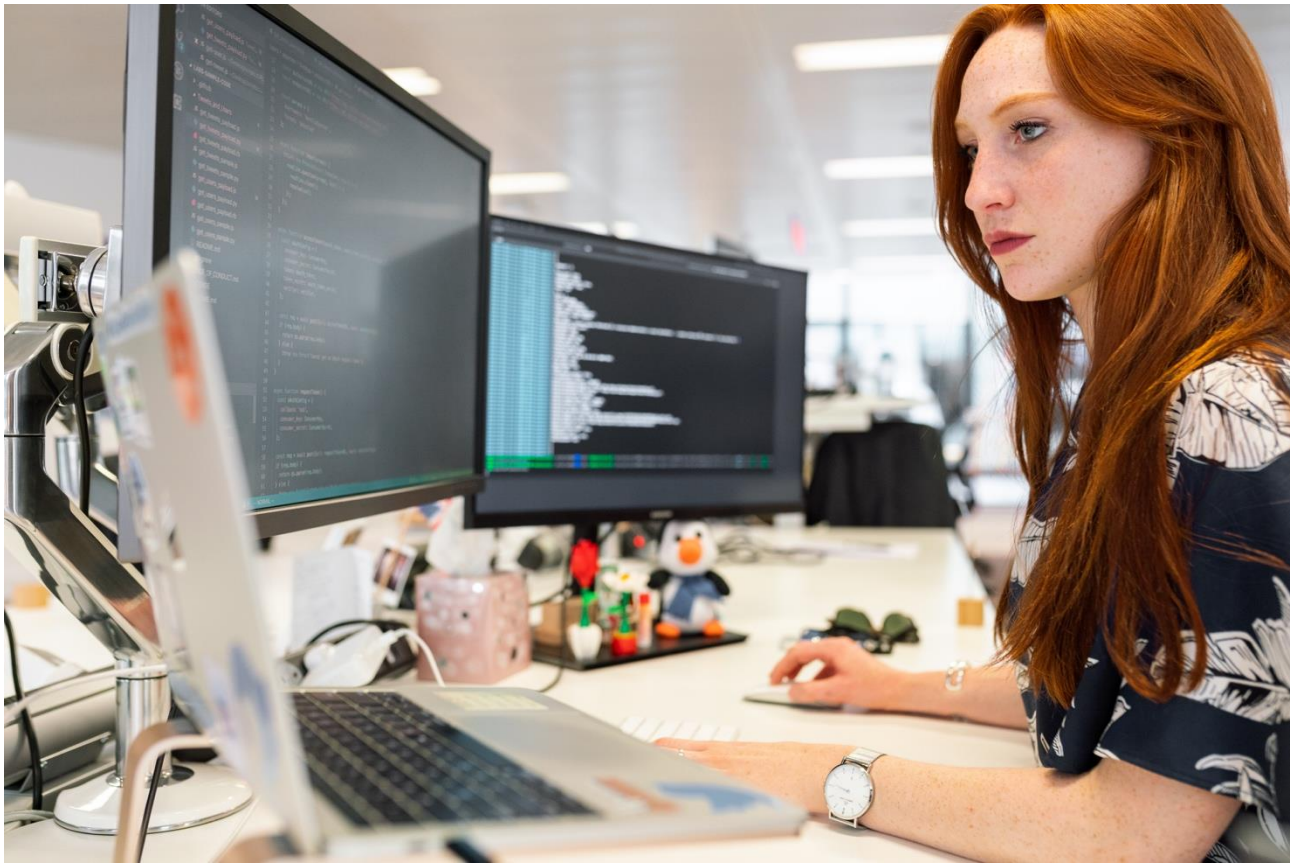
While computer science deals with high-level theoretical ideas, almost every aspect of modern life relies on coding. Every application on our phone, tablet, or computer uses computer languages to run.

Source: https://pixabay.com/photos/code-coding-computer-data-1839406/

Other digital systems like smart TVs and calculators use it too. Virtually every new car uses it to control everything from air conditioning systems to fuel injectors. Cities employ computers to operate traffic signals.

Systems that used to be analog are now streamlined using computerized systems. This allows engineers to build a more efficient and less expensive system, structure, and machine. Further, some of the most advanced technical fields (such as artificial intelligence and machine learning) use coding.[1]

2.2 Computer programmer



[1] Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. https://careerkarma.com/blog/what-is-coding-used-for/

Source: https://unsplash.com/photos/64YrPKiguAE

As mentioned before a computer programmer is the person responsible to create instructions for a computer to execute by writing and testing code that enables applications and software programs to operate successfully.
A computer programmer can also be referred to as a programmer, coder, developer, or software engineer. Also, the term is often used to refer to a stand-alone software developer, mobile applications developer, Web developer, software analyst, embedded firmware developer, and so on.

A computer programmer is a skilled professional who codes, tests, debugs, and maintains the comprehensive instructions known as computer programs that computers should follow to execute their specific functions[2].

Some of the common tasks a computer programmer has to master comprise:

- Testing software performance.
- Resolving computer software problems.
- Modifying software programs to improve performance.
- Writing computer programming code.
- Collaborating with others to resolve information technology issues.

It is common to classify computer programmers in two types: systems programmers and application programmers.

[2] Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com. https://www.techopedia.com/definition/6589/computer-programmer

Both the roles code. The difference between the two is that application programmers perform coding to manage a particular task, such as coding a program to monitor inventory within a company.

On the other hand, systems programmers code programs to maintain and control system software, including database management systems and operating systems (OSs).

2.1.1 Programming languages

Just like human-based languages, computers also have their own languages. Indeed, multiple computer programming languages exist for programmers to use to communicate with a computer to perform a set of specific tasks.
The portion of the language that a computer can understand is called a "binary." Translating programming language into binary is known as "compiling." Every programming language has its own distinct features. However, sometimes there are similarities between them[3].
Some of the most common computer programming languages are:

- **C**
- **SQL**
- **PHYTON**
- **JAVA**
- **R**
- **HTML[4]**

[3] Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. https://www.computerscience.org/resources/computer-programming-languages/
[4] Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. https://careerkarma.com/blog/top-programming-languages-2021/

Source: https://pixabay.com/illustrations/programming-languages-icon-898961/

Many programming languages are relatively simple, but they do different things. For example, one of the most popular languages, JavaScript, is used primarily for web pages and front-end development. On the other hand, Python is used for both complete software programs and websites.

Whichever a software developer chooses is usually up to their discretion, as more than one programming language can often work for the same purpose.
Knowing one of more of these programming languages is essential if we want to achieve a career not only as a computer programmer but also as a software engineer or every job in the data science field that requires high tech knowledge.

2.1.2 Programming process

The main objective of programming is generally to create a solution to resolve a problem. The problem can be something big on an international scale or simply something small to relieve us from boredom.

Source: https://pixabay.com/illustrations/programming-languages-icon-898961/

Developing a program involves steps similar to any problem-solving task. There are four main steps in the programming process:

- Defining the problem: at this stage you're trying to identify the problem. However, it would be wise to say that what you're really trying to identify is the solution of the problem, since you're trying to define what you want to achieve. The first thing to do is to analyse the requirements. Indeed, you need to find out what your program will be required to do.

The second thing to do then is to observe the list of requirements and decide exactly what your solution should do to fulfil them. Sometimes a single problem might require multiple solutions, so it's important to figure out what is the best solution for each problem.

Thus, you're trying to specify, once again, what solution you should adopt for your problem.

- Planning and designing the solution: this is generally the hardest task of all because it requires you to figure out how you will turn those specifications and requirements into a working and functional program.

  Designing a solution means describing in a higher level all the steps that the computer needs to follow in order to function properly or accomplish a specific task.

  This task is the hardest for different reasons: for example you might need to learn a bit more about the functions of your computer and your chosen programming languages to see what things can be easy or difficult; also as mentioned before a problem can require different solutions and once a solution is identified, it's important to analyse all its strengths and weaknesses and make the right choice.

  However, once the hard part is done, your design part should be completed. That means you have a very clear idea of how the computer is going to satisfy your specification, meeting your requirements, and finally solving your original problem.

- Coding the program: In this task you will have to instruct your computer how to perform your design. This phase includes 3 stages:
  - Coding: Coding means translating your design into a functional program, using a programming language. This stage practically involves

you sitting down and start typing codes on your computer. Just like an essay, coding a programme requires you to include things like titles, content page, introduction and references. Once you're done with your coding you need to submit it to the computer to see how the computer reacts and what makes of it.

- Compiling: The compiling phase requires you to turn the written programme in a language that the computer understands, hence the binary machine code. There are some programs called compilers that can do this for you. However, you have to be very precise when writing your program in a compiler, because they will detect every minimal mistake.
- Debugging: as mentioned before the compiler will detect every single mistake made in the program. Don't worry, that is a very common thing. That's why this stage is in place. Once you identify all the bugs in your program, thanks to the help of the compilers, look again at the original program, identify the mistakes, correct the code and recompile it. This cycle of code -> compile -> debug will often be repeated many times before the compiler is pleased with it and you will finally have a program that works. This might take a while.

  You can also decide just to write small sections of your program, compile them and debug them, rather than focusing on the whole program all at once.

- Testing the program: The final part of the programming process involves testing your creation and check that it works exactly the way you wanted it to. I know you might think that the compiler has already given you the correct program, however that doesn't mean the correct code can solve what your original problem was. That is left for you to check.

If eventual mistakes are found the only option is for you to revisit the program, fix the mistake by changing the code and re compile it. Remember to be careful when changing something in the code, because changing a small thing might affect other things in the whole program. This phase can be compared to another debugging phase.

Finally, after compiling again, fixing the mistakes, debugging and testing, and after making sure your program works according to your requirements and specifications, you will have a solution for your problem![5]

2.1.3 Programming principles

At this point we know that one of the main features of computer programming is to write codes. Whether you generally work alone or within a team, remember that your codes have to be easy to read and to maintain for other people.

That is why some basic programming principles that every programmer should know to write easy codes, with clean variables and able to stand strong even after testing and any modifications.

Some of these principles are:

- **KISS: Keep It Simple, Stupid: The KISS principle is one of the most important principles to live by in the programming world. It means you should be writing code as simple as possible. You don't have to try and show off with advanced code. If you can write it in one line, use one line. And remember that some codes need to be revisited after months, so keep that in mind and make it simple.**

[5] The Programming Process. (2020). Cs.Bham.Ac.Uk. https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html

- **DRY: Don't Repeat Yourself: This principle helps you remember a very common mistake in coding, hence repetitions. Avoid any duplicating of data and logic.**
- **Open/Closed: This principle makes sure you remember to keep your codes open for extensions but closed for modifications. This rule comes in handy when you're providing a library of services open for others to use. In very basic words, if someone else modifies your code, the code will break or something else in the code is affected and won't work as it normally should. That is why you should only release codes that prevent direct modification and encourage extension**
- **Single responsibility: This principle states that every class or module in a program should only provide one specific functionality. That is why this way the code is kept simple and also avoids a more complicated debugging process; Secondly, it becomes more difficult to create additional functionality for a specific module.**
- **YAGNI: You Aren't Going to Need It: This principle asserts that you should never try and write codes for a functionality you might need in the future. That would mean you're trying to solve a problem that doesn't exist.**

2.1.4 How can you become a computer programmer?

Being a computer programmer can help you find a job in the computer systems design and related services industry. However, many computer science fields require the figure of a computer programmer.

These include for example:

- **Software engineering**
- **Software development**
- **Computer engineering**
- **Computer graphics**
- **Artificial Intelligence**

Generally, it is required for computer programmers to have a university degree in data and computer science, which can give you the foundations for the beginning of your career. However, many programmers can also be self-taught enthusiasts filled with a persistent interest in programming.

Whether you decide to start your career with a university degree or as a self-enthusiast, the important thing is improving your knowledge on the programming languages, be interested in computer programming and also keep being informed and never stop learning, because computer programming is a constant up to date discipline.

Source: https://pixabay.com/photos/scrabble-board-game-game-4370255/

Another important thing that many computer programming job positions might require are professional and non-profit professional certifications available.
Some of them include:

- **CISCO – Certified Network Associate, Certified Network Profession Routing and Switching, Certified Network Associate Security Credential**
- **Microsoft – Certified Solution Developer for Web Applications, Certified Solutions Associate Windows Server**
- **Professional Associations – Software Development Associate Certification, Comptia's Security+, Comptia's A+ Certification, Comptia's Linux+**

- **Nonprofit – Certified Information Systems Security Professional, Certified Information Security Manager, Certified Secure Software Lifecycle Professional Credential6**

3. Assessment

3.1 Knowledge assessment

Quiz-like assessment based on the main content. Please mark the correct answer with bold when required. Include 10 questions for your module. Increase gradually the level of difficulty.

Question 1: The computer programmer is the professional figure responsible for writing codes, or *coding*, that instruct how a computer, application or software program should perform.
[True] [False]

Question 2: The coding process can only make a computer work.
[True][False]

Question 3: What tasks a computer programmer has to master?

[Resolving computer software problems] [Modifying software programs to improve performance] [Writing computer programming code] [Testing software performance] [All of the above]

Question 4: The portion of the language that a computer can understand is called a "binary." How is called the process of translating programming language into binary?

[6] Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming

[Coding] [Compiling] [Debugging]

Question 5: What is the right order of the programming process tasks?

[Coding the program, Testing the program, Defining the problem, Planning and designing the solution]
[Planning and designing the solution, Defining the problem, Coding the program, Testing the program]
[Defining the problem, Planning and designing the solution, Coding the program, Testing the program]

Question 6: What are the three stages of the Coding phase within the Programming process?
[Testing] [Coding] [Systemising] [Compiling] [Debugging]

Question 7: What do programming principles remind programmers to do when coding?
[to use codes that other programmers cannot modify in the future] [to use complex codes that enhance the functions] [to write easy codes] [to use clean variables that stand strong after testing]

Question 8 (matching): Match the terms with their definitions.

Binary: A base-2 number system invented by Gottfried Leibniz that's made up of only two numbers or digits: 0 (zero) and 1 (one).

ICT: This word refers to technologies that provide access to information through telecommunications.

Machine Learning: a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy

Analog: Adjective that refers to a mechanism or device in which information is represented by continuously variable physical quantities. It's opposed to digital.

Question 9 (matching): Match the concepts with their explanations.

Computer programmer: person responsible to create instructions for a computer to execute by writing and testing code that enables applications and software programs to operate successfully.

Compiling: process of translating programming language into binary.

Debugging: Process that involves identifying all the bugs in the program, thanks to the help of the compilers, review the original program, identify the mistakes, correct the code and recompile it.

Coding: process of translating the design into a functional program, using a programming language.

Testing: process that involves reviewing the program to check that it works exactly the way you wanted it to.

3.2 Skills assessment

Analytical thinking is an important trait in a programmer. It is also referred to as analytical reasoning, abstract thinking, or critical thinking. People that can think logically are able to analyse problems and devise solutions. This is not only valuable when developing programs but is vital to any situation that requires rational thought. People who are logical:

- **analyse information or resources related to a task**
- **carefully observe what is happening**
- **study information objectively to determine if it is relevant or true**
- **focus on facts not emotions**
- **develop solutions to problems based on facts**
- **outline ideas clearly by breaking them down into parts**
- **pay attention to details**
- **test the effectiveness of a solution and make revisions**

Many exercises and activities exist to practice this skill. Some of them are very simple, like reading books, playing brain games or joining a debate club to keep your brain stimulated.
In the case of a programmer an easy way to improve your analytical thinking skills is to follow and keep in mind this 5-step approach guide:

1. **Analyse the Problem: includes gathering information, examining resources, and determining gaps in skills or knowledge;**
2. **Formulate a Plan: brainstorm ideas and organise your thoughts into a plan;**

3. Develop Code to Solve the Problem: start writing the code (a knowledge of computer languages is essential);
4. Evaluate the Solution and Revise the Code: Review the solution and identify areas for improvement, if needed;
5. Justify Decisions: present evidence that explains why the program is an acceptable solution.

More information here: https://www.technokids.com/blog/teaching-strategies/its-easy-to-improve-logical-thinking-with-programming/

4. References

Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming

Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. https://careerkarma.com/blog/what-is-coding-used-for/

Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. https://www.computerscience.org/resources/computer-programming-languages/

Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. https://careerkarma.com/blog/top-programming-languages-2021/

Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com.

https://www.techopedia.com/definition/6589/computer-programmer

The Programming Process. (2020). Cs.Bham.Ac.Uk.

https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html