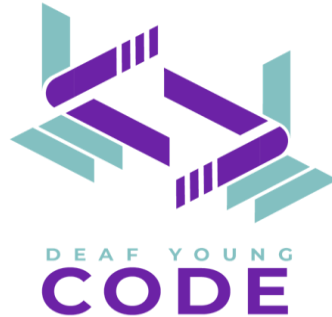




2023-2-IT03-KA220-YOU-000179130

worksheets for module 3

introduction to programming

use as individual work or team work



Co-funded by
the European Union

Worksheet – Introduction to Programming

Module: Introduction to Programming

Duration: 90 minutes

Level: Beginner

Target group: Deaf youth learning essential digital literacy

Teacher's Note:

Encourage participants to work in pairs. Use real or simulated pseudocode or email examples and allow hands-on device use for engagement.

Warm-up

Quick-Check (plenum, whole group)

Question

1. Coding = writing instructions, Coding is a part of programming.

Programming is more: it means the whole process to create a digital product.

yes ☐

no ☐

2. Coding and programming have the same meaning. There is no difference.

yes ☐

no ☐

Part 1 — Importance in today's digital world

Brainstorming:

Your **everyday life**: Think of all the applications you use and that need programming.

Examples: pocket calculator, traffic lights, alarm clock with light and snooze function

1) Write or draw at least 5 more applications.

2) Group-Discussion

Present your examples and discuss with others.

Part 2 – Key Elements – basic concepts

Following you see the important key elements:

2.1 Algorithms

2.2 Data Structures, Variables and data types

2.3 Control Structures, Control flow

2.4 Basic operators and expressions

2.5 Debugging

2.1 Algorithms

= Step-by-step guides for solving problems, like a recipe for cooking.

Exercise: Check your refrigerator

Develop a sorting algorithm for the refrigerator with the rule 'if expired, throw away'. Test with real products and adjust for efficiency.

Important: Products with a red star have expired and will be removed.



Let's start:

Milk > pack 1 > at the refrigerator side door, bottom compartment

[illegible]

2.2 Data structure, Variables and data types Methods for organizing and storing data, such as lists or dictionaries Examples: numbers, text.

2.2 Data structure, Variables and data types Methods for organizing and storing data, such as lists or dictionaries Examples: numbers, text.

2.2 Data structure, Variables and data types Methods for organizing and storing data, such as lists or dictionaries Examples: numbers, text.

Exercise: Apple cake recipe

160 g butter - 150 g sugar - 1 packet vanilla sugar - 3 eggs - 270 g plain flour - 3 tsp
baking powder - 1 pinch salt - 4 apples - 4 tbsp milk - 50 g almonds

Think about a data structure in the form of a table and enter everything correctly.

Each value needs to be assigned.

[illegible]

2.3 Control Structures, Control flow, if-else-flow

if-else: let the program make decisions.

„if" a condition is true > do one thing.

"else" > do something different.

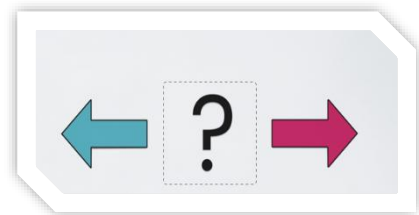
Example: Do I need a jacket today?

Build programme flow: 'If it's cold, I'll put on a jacket, otherwise I won't.'

Condition: 'Is it cold?'

If the answer is 'yes' → put on a jacket.

If the answer is 'no' → don't put on a jacket.



Example: Write an if-else loop with the function of a traffic light.

.....

.....

.....

.....

.....

.....



2.4 operators and expressions

Reusable sections of code that perform specific tasks.

Operators and expressions can be explained well with 'calculating and comparing', and 'reusable sections' such as small helper functions.

You will be familiar with some operators from your maths lessons.

Examples:

+ (plus): Adds two numbers, e.g. $5 + 3$ equals 8.

- (minus): Subtracts, e.g. $10 - 4$ equals 6.

* (Multiply): Multiplies, e.g. $4 * 3$ equals 12.

/ (Divide): Divides, e.g. $12 / 3$ equals 4.

Exercise:

Imagine you want to calculate how many hours you spend on the internet in total.
Calculate this for one week.

- 1) Which operators did you use?
- 2) What is your result?

.....

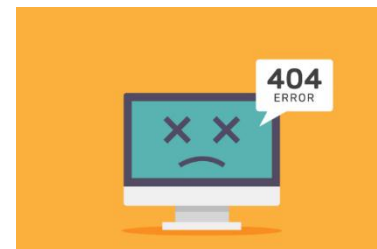
.....

2.5 Debugging

Debugging means finding and fixing mistakes in a program
so that it does what you want.

Excercise: Drawing Game

Pairwork: test – improve – test



One Participant is the “programmer” and the other is the “rovot”.

The programmer gives very precise step-by-step instructions how to draw a picture.

When something goes wrong or looks funny, stop and say: **“There is a bug in our program.”**

Then the programmer must change the instruction so the robot can do the task correctly.

Examples for drawings:



Part 3 — Quick Quiz

Mark the correct answer:

What is a Program ?

- A) A type of hardware used in computers.
- B) A game played on a computer.
- C) A set of instructions that a computer follows to perform a task.
- D) A device that stores data.

2. Which of the following is an example of a variable in programming?

- A) A command to print text on the screen.
- B) A storage location that holds data, like a number or text.
- C) A message that explains what the code does.
- D) A type of programming language.

3. What does a loop do in a program?

- A) Checks if a condition is true or false.
- B) Repeats a set of instructions multiple times.
- C) Stores data for later use.
- D) Sends data to another computer.

4. Why is debugging important in coding?

- A) To add new features to the program.
- B) To identify and fix errors or bugs in the code.
- C) To make the program run faster.
- D) To translate code into different languages

Part 4 — Reflection

- Which digital skill was new for you? _____
- Which task was the most fun? _____
- What would you like to practice more? _____