



AGH University of Krakow | Faculty of Mechanical Engineering and Robotics  
| Department of Robotics and Mechatronics

MECHATRONIC ENGINEERING | I cycle | semester VI | 2025/2026

## Object oriented programming and software engineering

### Instruction I:

### *Git and C++ basics*

#### You will learn:

How to use basic version control with Git, including creating repositories, tracking changes, creating a commit, working with branches, and collaborating through a remote repository. You will also learn how to write, compile, and run simple C++ programs, use fundamental language constructs such as variables, input/output, conditionals, loops, and functions small programs clearly and correctly. These skills will form the foundation for more advanced programming tasks and collaborative software development later in the semester.

#### Course supervisor:

dr inż. Lucjan Miękina, [miekina@agh.edu.pl](mailto:miekina@agh.edu.pl)

#### Instruction author:

mgr inż. Joanna Koszyk, [jkoszyk@agh.edu.pl](mailto:jkoszyk@agh.edu.pl)

## 1. Initial information

**Git** is a distributed version control system, which means that every repository contains the same information. In most cases, there is a central server used to store a shared repository, but each cloned repository is a full copy of that repository. Version control systems are not limited to working with text files, they can handle files of any type, including binary files.

**C++** is a general-purpose programming language used for developing system software, applications, embedded systems, and high-performance programs. It is a compiled language, which means that the source code must be translated into machine code by a compiler before it can be executed.

## 2. Theoretical content

**Git:**

**git init** – creates a new, empty repository in the current directory

**git clone** – clones a remote repository into a local directory

**git add** – adds a file to the index

**git add .** – adds all modified files

**git commit** – creates a commit

**git commit -m "message"** – creates a commit with a message

**git commit -am "message"** – adds changes in tracked files and creates a commit

**git status** – shows the current status of files

**git remote add origin [address]** – sets the remote repository as origin

**git fetch** – downloads the latest changes from the remote repository but does not merge them

**git pull** – downloads and merges changes from the remote repository

**git push** – sends commits to the remote repository

**git branch** – shows the list of branches

**git branch [new\_branch\_name]** – creates a new branch

**git checkout** – switches to another branch

**git diff** – shows changes between commits, the commit and the working directory, etc.

**git merge** – merges a branch into the current branch

**C++:**

Basic data types:

**int** – integer numbers

**double** – real numbers

**char** – single characters

**std::string** – text

Input and output:

**std::cout** – output to console

**std::cin** – input from console

Conditional statements

**if**

**if-else**

**nested conditions**

Loops

**for loop**

**while loop**

**do-while loop**

### 3. Tasks

#### TASK 1.

Create an account at: <https://gitlab.com/> In the next step, you need to log in (identify the person making the changes). To do this, you should provide your details (username and email address) in the config. For the purposes of the course, you must use an email address in the student domain.

```
C:\Users\PC>git config --global user.name jkoszyk
C:\Users\PC>git config --global user.email jkoszyk@agh.edu.pl
C:\Users\PC>git config --global -l
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=jkoszyk
user.email=jkoszyk@agh.edu.pl
C:\Users\PC>
```

Create a repository:

```
C:\Users\PC\Downloads>mkdir poio-proj2
C:\Users\PC\Downloads>cd poio-proj2
C:\Users\PC\Downloads\poio-proj2>git init
Initialized empty Git repository in C:/Users/PC/Downloads/poio-proj2/.git/
C:\Users\PC\Downloads\poio-proj2>echo "# project1" >> README.md
C:\Users\PC\Downloads\poio-proj2>git add README.md
C:\Users\PC\Downloads\poio-proj2>git commit -m "first commit"
[master (root-commit) 311e3a4] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
C:\Users\PC\Downloads\poio-proj2>git branch -M main
C:\Users\PC\Downloads\poio-proj2>git remote add origin https://gitlab.com/jkoszyk-agh-group/project2
C:\Users\PC\Downloads\poio-proj2>git push -u origin main
warning: redirecting to https://gitlab.com/jkoszyk-agh-group/project2.git/
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 222 bytes | 222.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote:
remote: The private project jkoszyk-agh-group/project2 was successfully created.
remote:
remote: To configure the remote, run:
remote:   git remote add origin https://gitlab.com/jkoszyk-agh-group/project2.git
remote:
remote: To view the project, visit:
remote:   https://gitlab.com/jkoszyk-agh-group/project2
remote:
remote:
remote: To https://gitlab.com/jkoszyk-agh-group/project2
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
C:\Users\PC\Downloads\poio-proj2>
```

## TASK 2.

Create your first .cpp program and add the program in the repository. Create another branch.

```
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

Compile and run:

```
g++ hello.cpp -o hello
hello.exe
```

## TASK 3.

Write a C++ program that calculates the robot's final position starting from (0,0). The user enters movement directions, and the program updates the robot's position based on the steps taken.