

**Poznan University of Technology**  
**Object Oriented Programming**

Jakub Piotr Hamerliński

# Object Oriented Programming

## Agenda

01. About me
02. Topics
03. How to pass?
04. Sources
05. Contact and Q&A
06. Tasks
07. "Homework"



**About me**

## About me

My name is Jakub Piotr Hamerliński, M.Eng.  
I'm a DevOps engineer passionate about  
cybersecurity and cryptography.

Pronouns: he/him/his.

[My GitHub](#) | [My LinkedIn](#)



# Topics

# Object Oriented Programming

## Topics

01. Introduction
02. Git usage and open source
03. Language basics
04. Class, attribute
05. Method, object
06. Constructor
07. Interface
08. Exception
09. Testing
10. Decorator
11. RAIL
12. Other design patterns

**How to pass?**

# Object Oriented Programming

## How to pass?

01. Exercises during classes and work in the classroom + tests/quiz.
02. Project.
03. Extra points for open source contribution.

Lessons typically will start with topic introduction. Then students will perform small and simple tasks which will be reviewed and graded before end of class. After teacher's approval, students must commit their solution.

$$\mathbb{X} = 0.5 \times \mathbb{E} + 0.5 \times \mathbb{P}$$

where  $\mathbb{E}$  means average from excercises,  
and  $\mathbb{P}$  means grade from the project

**Each of the components of the grade must be positive.**



# Object Oriented Programming

## How to pass? - Project

Project must obey all rules presented up to (including) Decorator design pattern.

Student must prepare 5 min presentation (with slides) which will answer following questions:

01. What have you created?
02. Did you have any problems?
03. Did you learn anything new?
04. What could be improved?

Student proposes project's topic which must be accepted by teacher.

Students from groups **1, 2, and 3** will present their projects on **29th May**. Deadline: **2023-05-28 23:59:59**

Students from group **4** will present their projects on **30th May**. Deadline: **2023-05-29 23:59:59**

All code and presentation must be committed to your public repository to folder **project** before deadline.

# Object Oriented Programming

## How to pass?

2 absences from classes allowed. Each subsequent one must be made up.

# Sources

# Object Oriented Programming

## Sources

Great books about OOP:

01. David West, *Object Thinking*, Microsoft Press, 2004
02. Yegor Bugayenko, *Elegant Objects (Volume 1)*, CreateSpace, 2016
03. Yegor Bugayenko, *Elegant Objects (Volume 2)*, CreateSpace, 2017

My slides (Work in progress): <https://github.com/hamerlinski/slides-oop>

# Contact and Q&A

# Object Oriented Programming

## Contact and Q&A

Please contact me using [jakub.hamerlinski@cs.put.poznan.pl](mailto:jakub.hamerlinski@cs.put.poznan.pl)

Questions?

# Tasks



# Object Oriented Programming

## Tasks

01. Create GitHub account (if you don't have it yet) and repository.
02. Fill form.

# Object Oriented Programming

## Task - GitHub account creation + repo

1.1 Create GitHub account using following guide: [Create a GitHub account to use with Visual Studio](#)

1.2 Create repository `oop-put-course` from [the template](#) using following guide:


[Creating a repository from a template](#)

### Create a new repository from oop-put-course-template


The new repository will start with the same files and folders as [hamerlinski/ooop-put-course-template](#).

Owner \*

Repository name \*

 hamerlinski

/

oop-put-course 

Great repository names are short and memorable. Need inspiration? How about [cuddly-computing-machine?](#)

Description (optional)

☒ Public


Anyone on the internet can see this repository. You choose who can commit.

☐ Private


You choose who can see and commit to this repository.

☐ Include all branches

Copy all branches from hamerlinski/ooop-put-course-template and not just main.

 You are creating a public repository in your personal account.

Create repository from template

 © 2023 GitHub, Inc.

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

# Object Oriented Programming

## Task - form

02. Fill form: [tinyurl.com/4wab33bw](https://tinyurl.com/4wab33bw)



**"Homework"**

# Object Oriented Programming

## "Homework"

Prepare for next lesson by reading and analyzing slides oo-cpp-basics available at

<https://github.com/hamerlinski/slides-ooop/tree/feature/refactor-slides-for-bioinformatics/slides/oo-cpp-basics>

# Thank you

Feel free to reach me via [LinkedIn](#)

***Fin***