University of Zagreb

Faculty of Electrical Engineering and Computing

Department of Applied Computing

**Introduction to programming**

**A guide for students**

Zagreb, 2023.

**Contents**

[1 Introduction 1](#_Toc52293287)

[2 Forms of teaching 2](#_Toc52293288)

[2.1 Lectures 2](#_Toc52293289)

[2.2 Laboratory exercises with knowledge tests 2](#_Toc52293290)

[2.3 Consultations 2](#_Toc52293291)

[2.4 Knowledge tests 2](#_Toc52293292)

[3 Workload, grading elements and grades 3](#_Toc52293293)

[3.1 Workload 3](#_Toc52293294)

[4 Conditions for obtaining a passing grade 3](#_Toc52293295)

[5 Scoring thresholds for determining final grades 4](#_Toc52293296)

[6 Other materials 5](#_Toc52293297)

[6.1 Recommended reading 5](#_Toc52293298)

[6.2 Instructions 5](#_Toc52293299)

[6.3 Official reminder 5](#_Toc52293300)

[6.4 Lecturers 6](#_Toc52293301)

# Introduction

The aim of the course is to adopt the basic concepts of programming and gain the ability to create less complex programs in the programming language C.

Lecture slides, instructions, recommended literature, official reminders, and other materials are published through the **IPRO2023** team on the *Microsoft Teams* platform.

Also, in case of need, students will receive the notification by e-mail, therefore you should check your email inbox @fer.hr.

# Forms of teaching

## Lectures

The lectured present theoretical assumptions and explains concepts that are supported by numerous explanations and examples. There will be short tests given during most lectures which will count towards the final grade.

Lectures take place in two cycles. After the first cycle, midterm exams are held, and after the second

cycle final exams.

* lectures in the 1st cycle, October 2nd - November 15th, 2023.
* lectures in the 2nd cycle, December 4th, 2023 - January 24th, 2024.

During the lectures, the planned material is presented with the help of Powerpoint slides, solve tasks, discuss, conduct ad hoc surveys and knowledge tests which can be used by students to determine knowledge and self-assessment

* slides that will be presented at lectures are available through the **IPRO2023** team on the *Microsoft Teams* platform
* the content of the slides, since they are published in advance, don’t need to be copied, but it is advised to take your own notes during lectures
* actively participate in lectures: solve tasks assigned during lectures, ask teacher for additional clarifications
* just following lectures and/or reading presentations will certainly not be enough to pass the exam, so pay close attention to the content of the *Independent work* chapter

## Independent work

The course is performed with a load of **7** ECTS credits, which means that the total student engagement required to successfully master the material is estimated at approximately **7** x 30 = 210 hours. It is envisaged that

* 50 hours will be spent on attending classes,
* 20 hours will be spent participating in laboratory exercises and attending exams,
* the remaining time, i.e. at least 140 hours, must be spent on independent work.

It should be noted that 140 hours**,** when distributed on all teaching weeks**,** equal to an average of approximately 10 hours of independent work per week. Due to the layered approach in the processing of materials, ***it is very important to work continuously*,** every week. Campaign work (e.g. "I won’t do anything for three weeks and then I’ll make up for it all at once") on this subject will almost certainly end in very poor results.

Independent work includes:

* Learning
  + Slides from the lectures
  + Your own notes
  + Reading other literature (see chapter *Recommended reading*)
  + Internet
* Solving tasks
  + solving a large number of tasks on your own computer acquires the necessary skills (simple algorithm design, code writing, translation, error detection, testing) which are tested in laboratory exercises and written tests (midterm, final exam, exam).
  + It is therefore of the utmost importance:
  + solve published assignments for exercises in addition to lectures (published at the same time as lectures)
  + test examples of programs and excerpts from lectures, modify these examples yourself, notice the effects of modifications (you can copy the program code of examples from lectures from PDF versions of published lectures)
* Reading other literature (see chapter *Other materials - Recommended reading*)
* Browsing content on the Internet for additional explanations and examples

## Laboratory exercises with knowledge tests

All students perform laboratory exercises in a FER lab, in precisely prescribed terms:

* four exercises in the 1st cycle (i.e., before the midterm exam) and four exercises in the 2nd cycle
* the duration of one laboratory exercise is approximately 110 minutes
* laboratory exercises must be completed by the student in the specified time
* there is no compensation for laboratory exercises in case of absence
* the laboratory exercise consists of two parts:
  + solving tasks in which the correct one must be chosen from among several offered answers ("multiple choice questions"). A correct answer obtains positive points, an incorrect answer obtains negative points. An unanswered question does not earn negative points.
  + creation of one or more programs (algorithm design, coding, testing). In these assignments there are no negative points.

Each student works independently on the faculty computer, but under the supervision of the teachers on duty. The student has at his disposal the same tools that he uses to solve tasks on his own computer:

* Notepad and Notepad++ editors, gcc compiler used via command line of the operating system
* VSCode development environment

**IMPORTANT:** regardless of which of the options the student chooses, knowledge of elementary program editing with the Notepad/Notepad++ editor, as well as compiling and running the program from the command line is mandatory!

Laboratory exercise includes the development of one or more programs (algorithm design, coding, testing). The exact schedule by dates, topics and number of program tasks per exercise, and the maximum number of points that can be obtained in each exercise will be published on time, at least a week before the exercises, on **Teams**. Students will submit solutions to their tasks during laboratory exercises via the Edgar system. Instructions for that will be available on time.

## Consultations

Students have the opportunity to consult with the teacher. Consultations will take place through the MS Teams platform, at times agreed with the teacher.

* the teacher will be happy to additionally explain the material to you during the consultation, but will not teach it again
* consultation should be requested as soon as ambiguities arise that you are unable to resolve on your own or with the help of colleagues. Do not postpone the consultation until just before the midterm exam, final exam or exam - at that point it will most likely be too late.

## Knowledge tests

During the continuous assessment process, two written exams are taken: midterm exam and final exam.

* exact time is TBD
* part of the tasks can be in the form of questions, part of the tasks can be based on choosing the correct answer, but most of the tasks are related to solving a given problem in the programming language C.
* tasks are solved on a computer at FER
* exams are graded by Edgar, but these points do not have to be final; final points are assigned by teachers after reviewing student solutions
* there is no compensation for midterm exams and final exams in case of absence

Students who fail to pass the course during the continuous monitoring of classes, can take one or more classical exams.

* exact time is TBD
* classical exams are similar to the midterm exam and the final exam, but, as a rule, are more extensive

# Conditions for obtaining a passing grade

## During the continuous assessment

The total number of points gained during the continuous assessment is obtained by adding the points gained at lectures, points gained in laboratory exercises and points gained in the midterm exam and the final exam:

* total number of points = number of points at lectures + number of lab points + number of points on intermediate exam + number of points on final exam

|  |  |  |
| --- | --- | --- |
| Based on continuous assessment | Maximum  points | Threshold  points |
| Lectures | 10 | there is no threshold |
| Laboratory exercises | 30 | there is no threshold |
| Midterm exam | 25 | there is no threshold |
| Final exam | 35 | **14** |

Conditions for obtaining a passing grade based on continuous assessment:

* achieved a threshold of at least 14 (out of 35 possible) points on the final exam ***and***
* total points ³ 50

## On classical exams

For students who did not pass the exam during the continuous assessment, the total number of points is obtained by summing the points obtained in laboratory exercises during the continuous assessment and points obtained in the written part of the exam

* total number of points = number of lab points + number of points on the exam

|  |  |  |
| --- | --- | --- |
| On classical exam | Maximum  points | Threshold  points |
| Laboratory exercises | 30 | there is no threshold |
| Exam | 70 | **35** |

Conditions for obtaining a passing grade in the exam period:

* achieved a threshold of at least 35 (out of 70 possible) points on the written exam **and**
* total points ³ 50

# Scoring thresholds for determining final grades

Students who meet the requirements for obtaining a passing grade based on continuous assessment or on the exam period receive a grade based on the following table:

|  |  |
| --- | --- |
| Total points earned | Grade |
| [87.5, 100] | Excellent - 5 |
| [75, 87.5) | Very good- 4 |
| [62.5, 75) | Good - 3 |
| [50, 62.5) | Sufficient - 2 |

# Other materials

## Recommended reading

* V. Mornar: Introduction to programming, working version of the textbook
  + The textbook covers the complete material of the subject, and even more than that. The textbook will be available on MS Teams as pdf. An integral part of the textbook is the folder *Examples* with about ninety programs that can be independently translated and run in the command line or in the development environment VSCode.
* K. N. King, C Programming: A Modern Approach, Second Edition, W. W. Norton & Company
  + An extensive work (805 pages) that has been accepted as a textbook at hundreds of universities around the world.
* B. W. Kernighan, D. M. Ritchie: The C Programming Language, Second Edition, Englewood Cliffs, NJ: Prentice Hall
  + A basic book whose authors are the creators of the C programming language from which many generations of programmers have learned the language.

## Instructions

* Instructions for compiling programs written in the programming language C
  + contain a detailed description of the software installation procedures and how to compile and execute the program. Based on these instructions, students must learn to use some of the simple editors (Notepad or Notepad ++) and compile programs using the operating system command line. In addition, students can use these instructions, if they wish, to install and learn the basics of using the integrated development environment VSCode, which greatly facilitates error detection because it has a simple interaction with a specialized program - a debugger.
* Instructions for using the character interface in Windows
  + basics of using the operating system's character interface (command line). Based on these instructions, students are expected to independently learn several operating system commands listed there: basics of positioning in file system folders, creating, renaming files and folders, redirecting inputs and outputs.
* Instructions for laboratory exercises and the Edgar system – will be published in time

## Official reminder (*cheat sheet*)

The official reminder can be printed on the paper and used freely in lectures and any knowledge test: laboratory exercises, midterm exam, final exam and classical exams. Nothing may be added to the copies of official reminders that you will use in knowledge tests.

* Official reminder for programming language C -
  + a very useful tool with information that teachers feel as unnecessary to memorize. It is recommended to use the official reminder both in lectures and during independent work, in order to get acquainted with where some information can be found, which facilitates the use of reminders in knowledge tests.
  + on a copy of the official reminder nothing should be added
  + this reminder will be available in digital form (pdf file) at the laboratory exercises in the Edgar system

## Lecturers

|  |
| --- |
| Lecturer |
| Prof. dr. sc. Vedran Mornar, vedran.mornar@fer.hr |

Assistant: Melita Fertalj, mag. Ing., melita.fertalj@fer.hr

If communicating via email, please start subject with **[IPRO2023]**.