Computer Programming

|  |  |
| --- | --- |
| **Course code** | *IT103* |
| **Compulsory in the programmes** | *Economics and Finance* |
| **Level of studies** | *Undergraduate* |
| **Number of credits** | *6 ECTS (48 in-class hours + 2 consultation hours + 2 exam hours, 110 individual work hours)* |
| **Course coordinator (title and name)** | *M.Sc. Oleg Mirzianov* |
| **Prerequisites** | *None* |
| **Language of instruction** | *English* |

**THE AIM OF THE COURSE:**

This ***intense*** course is based on the Harvard CS50x course curriculum - introduction to the intellectual enterprises of computer science and the art of programming. Through the course, students learn how to solve real-life problems in the most efficient way using algorithms. The course introduces problem-solving, which is inspired by the arts, humanities, social sciences, and economics especially. Course covers C, Python, SQL languages. At the end of the course students are expected to create a personal final project.

**MAPPING OF COURSE LEVEL LEARNING OUTCOMES (OBJECTIVES) WITH DEGREE LEVEL LEARNING OBJECTIVES (See Annex), ASSESSMENT AND TEACHING METHODS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course level learning outcomes (objectives)** | **Learning objectives for BSc in Business Management** | **Learning objectives for BSc in Social Science** | **Assessment methods** | **Teaching methods** |
| CLO1. To be able to process information and operate at multiple levels of abstraction. | BLO4.1. Students will be able to communicate reasonably in different settings according to target audience tasks and situations. | ELO4.1.Students will be able to communicate reasonably in different settings according to target audience tasks and situations | Coursework, midterm, final exam, final project. | Lecture, consultation, problem solving, homework, discussion. self-study |
| CLO2. To be able to decompose IT problems into parts and solve them efficiently. | BLO4.3. Students will be able to convey their ideas effectively in a written paper. | ELO4.3. Students will be able to convey their ideas effectively in a written paper | Coursework, midterm, final exam, final project. | Lecture, consultation, problem solving, homework, discussion. |
| CLO3. To be able to demonstrate proficiency in a software development environment. | BLO3.2. Students will be able to make decisions using appropriate IT tools. | ELO3.2. Students will be able to make decisions using appropriate IT tools | Coursework, final exam, final project. | Lecture, consultation, problem solving, homework, discussion. |
| CLO4. To be able to assess the correctness, design, and style of code. | BLO3.2. Students will be able to make decisions using appropriate IT tools. | ELO3.2. Students will be able to make decisions using appropriate IT tools | Coursework, midterm, final exam, final project. | Lecture, consultation, problem solving, homework, discussion. |
| CLO5. To be able to evaluate the project’s complexity and estimate required resources. | BLO1.2. Students will be able to conduct a contextual analysis to identify a problem associated with their discipline, to generate managerial options and propose viable solutions. | ELO3.1. Students will demonstrate proficiency in common business software packages | Final project, coursework. | Lecture, consultation, problem solving, homework, discussion. |

**ACADEMIC HONESTY AND INTEGRITY**

The ISM University of Management and Economics Code of Ethics, including cheating and plagiarism are fully applicable and will be strictly enforced in the course. Academic dishonesty, and cheating can and will lead to a report to the ISM Committee of Ethics. With regard to remote learning, ISM reminds students that they are expected to adhere and maintain the same academic honesty and integrity that they would in a classroom setting.

The course’s philosophy on academic honesty is best stated as “be reasonable.” The course recognizes that interactions with classmates and others can facilitate mastery of the course’s material. However, there remains a line between enlisting the help of another and submitting the work of another. This policy characterizes both sides of that line.

The essence of all work that you submit to this course must be your own. Collaboration on problem sets is not permitted except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code to others, but you may not view theirs, so long as you and they respect this policy’s other constraints. Collaboration on the course’s final exam and test is not permitted at all. Collaboration on the course’s final project is permitted to the extent prescribed by its specification.

|  |  |
| --- | --- |
| Reasonable | Not reasonable |
| Communicating with classmates about problem sets’ problems in English (or some other spoken language), and properly citing those discussions.  Discussing the course’s material with others in order to understand it better.  Helping a classmate identify a bug in their code at lectures, elsewhere, or even online, as by viewing, compiling, or running their code after you have submitted that portion of the problem set yourself. Add a citation to your own code of the help you provided and resubmit.  Incorporating a few lines of code that you find online or elsewhere into your own code, provided that those lines are not themselves solutions to assigned problems and that you cite the lines’ origins.  Reviewing past semesters’ tests and final exams and solutions thereto.  Sending or showing code that you’ve written to someone, possibly a classmate, so that he or she might help you identify and fix a bug, provided you properly cite the help.  Turning to the course’s heads for help or receiving help from the course’s heads during a final exam or test.  Turning to the web or elsewhere for instruction beyond the course’s own, for references, and for solutions to technical difficulties, but not for outright solutions to problem set’s problems or your own final project.  Whiteboarding solutions to problem sets with others using diagrams or pseudocode but not actual code. | Accessing a solution to some problem prior to (re-)submitting your own.  Accessing or attempting to access, without permission, an account not your own.  Asking a classmate to see their solution to a problem set’s problem before (re-)submitting your own.  Discovering but failing to disclose to the course’s head bugs in the course’s software that affect scores.  Decompiling, deobfuscating, or disassembling the staff’s solutions to problem sets.  Failing to cite (as with comments) the origins of code or techniques that you discover outside of the course’s own lessons and integrate into your own work, even while respecting this policy’s other constraints.  Giving or showing to a classmate a solution to a problem set’s problem when it is he or she, and not you, who is struggling to solve it.  Looking at another individual’s work during the final exam or test.  Manipulating or attempting to manipulate scores artificially, as by exploiting bugs or formulas in the course’s software.  Paying or offering to pay an individual for work that you may submit as (part of) your own.  Providing or making available solutions to problem sets to individuals who might take this course in the future.  Searching for or soliciting outright solutions to problem sets online or elsewhere.  Splitting a problem set’s workload with another individual and combining your work.  Submitting (after possibly modifying) the work of another individual beyond the few lines allowed herein.  Submitting the same or similar work to this course that you have submitted or will submit to another.  Submitting work to this course that you intend to use outside of the course (e.g., for a job) without prior approval from the course’s heads.  Turning to humans (besides the course’s heads) for help or receiving help from humans (besides the course’s heads) during the final exam or midterm.  Viewing another's solution to a problem set’s problem and basing your own solution on it. |

**COURSE OUTLINE**

|  |  |  |
| --- | --- | --- |
| **Topic** | **In-class hours** | **Readings** |
| Scratch  Problem solving  Inputs, Outputs  Representation  Unary, Binary, Decimal  Abstraction  ASCII, Unicode  RGB  Algorithms  Running Time  Pseudocode  Scratch   * Functions, Arguments, Return Values * Variables * Boolean Expressions, Conditions * Loops * Events * Threads | 4 | https://cs50.harvard.edu/college/2020/spring/weeks/0/ |
| C language - Basics  Linux  Command-Line Interface  Programming language   * Functions, Arguments, Return Values * Variables * Boolean Expressions, Conditions * Loops   Libraries, Header Files  Text Editors  Terminal Windows  Compiler  Types  Integer Overflow  Floating-Point Imprecision | 4 | https://cs50.harvard.edu/college/2020/spring/weeks/1/ |
| C language - Arrays  Preprocessing  Compiling  Assembling  Linking  Debugging  Arrays  Strings  Command-Line Arguments  Cryptography | 8 | https://cs50.harvard.edu/college/2020/spring/weeks/2/ |
| C language - Algorithms  Searching   * Linear Search * Binary Search   Sorting   * Bubble Sort * Selection Sort * Insertion Sort * Merge Sort   Asymptotic Notation   * O * Ω * Ɵ   Recursion | 8 | https://cs50.harvard.edu/college/2020/spring/weeks/3/ |
| Midterm | 2 | Review all previous material |
| Python   * Functions, Arguments, Return Values * Variables * Boolean Expressions, Conditions * Loops   Modules, Packages | 12 | <https://cs50.harvard.edu/college/2020/spring/weeks/6/> |
| SQL   * Tables * Types * Statements * Constraints * Indexes * Keywords, Functions * Transactions   Race Conditions  SQL Injection Attacks | 8 | <https://cs50.harvard.edu/college/2020/spring/weeks/7/> |
| Final project | 4 |  |
|  | **Total: 48 hours** |  |
| CONSULTATIONS | 2 |  |
| FINAL EXAM | 2 |  |

**DESCRIPTION AND GRADING CRITERIA OF EACH ASSIGNMENT**

|  |  |  |
| --- | --- | --- |
| **Type of an assignment** | **Hours** | **Course grade weights (%)** |
| Problems sets (6 assignments) | 50 | 40% |
| Labs (5 small tasks) | 6 | 10% |
| Midterm | 16 | 15% |
| Final exam | 16 | 15% |
| Final project | 24 | 20% |
| **Total:** | **112** | **100%** |

|  |  |
| --- | --- |
| **Problem set** | **Course grade** |
| Problem 0 | 0.4 |
| Problem 1 | 0.4 |
| Problem 2 | 0.8 |
| Problem 3 | 1.2 |
| Problem 5 | 0.8 |
| Problem 6 | 0.4 |
| **Total:** | **4** |

Problem sets and the final project are evaluated along axes of correctness and style, with correctness ordinarily counting for 90% of your score and style counting for 10%.

*Note*: active participation during the course might contribute max. 1.5 points to the final evaluation. Meaning, you might complete the course with mark 10 without attending the exam.

**Lateness**

Late submissions (of the problem sets and the final project’s milestones) will be penalized at a rate of 0.1% per minute:

* If you submit 10 minutes late, your score will be penalized 1%. Your score will thus be 99% of what it would have been if submitted on time.
* If you submit 60 minutes late, your score will be penalized 6%. Your score will thus be 94% of what it would have been if submitted on time.
* If you submit 1,000 minutes (just over 16 hours) late, your score will be penalized 100%. Your score will thus be effectively zeroed.

**RETAKE POLICY**

In case the final grade is less than five (not passed), students can be allowed to have one retake. Retake means: retake midterm and final exam. The evaluation of the problem sets and the final project will not be affected by the retake. The percentage of the final grade that can be affected by the retake is equal to 30% (15% midterm and 15% exam).

**ADDITIONAL REMARKS**

**Labs**

Covers the material from theoretical lectures and is organized as a small programming task.

**Midterm**Covers all the material which was presented before the midterm. Midterm form **i**s organized as a 2 hours programming task. This midterm applies **open-book rules.**

**Final exam**

Covers all the material which was presented during the course. Final exam is organized as a 2 hours programming task.

**Final project**

The final project is the final assignment of the course. Students get the opportunity to choose any topic they want. So long as your project draws upon the lessons of this course, the nature of your project is entirely up to you, albeit subject to the staff's approval. Students are asked to pick the idea they want and implement it with preferred technology in the way that it solves an actual problem, that impacts the campus, or that changes the world.

That being said, there are some provisos. You may implement your project in any programming language(s) as long as the teaching staff approves it. You are welcome to utilize any infrastructure, provided the staff ultimately has access to any hardware and software that your project requires. The final project length should be equivalent to at least 1.5 problem set. Final projects should be presented for the course heads in up to 3 min presentations in virtual or physical format. Overall implementation of the final project will be taken into account when grading the final project.

**REQUIRED READINGS**

https://cs50.harvard.edu/college/

**ADDITIONAL READINGS**

Optional:

1. *Hacker’s Delight*, Second Edition Henry S. Warren Jr. Pearson Education, 2013 ISBN 0-321-84268-5
2. *How Computers Work*, Tenth Edition Ron White Que Publishing, 2014 ISBN 0-7897-4984-X
3. *Programming in C*, Fourth Edition Stephen G. Kochan Pearson Education, 2015 ISBN 0-321-77641-0

**ANNEX**

**DEGREE LEVEL LEARNING OBJECTIVES**

**Learning objectives for the Bachelor of Business Management**

*Programmes:*

*International Business and Communication,*

*Business Management and Marketing, Finance,*

*Industrial Technology Management*

|  |  |
| --- | --- |
| **Learning Goals** | **Learning Objectives** |
| Students will be critical thinkers | BLO1.1. Students will be able to understand core concepts and methods in the business disciplines |
| BLO1.2. Students will be able to conduct a contextual analysis to identify a problem associated with their discipline, to generate managerial options and propose viable solutions |
| Students will be socially responsible in their related discipline | BLO2.1. Students will be knowledgeable about ethics and social responsibility |
| Students will be technology agile | BLO3.1. Students will demonstrate proficiency in common business software packages |
| BLO3.2. Students will be able to make decisions using appropriate IT tools |
| Students will be effective communicators | BLO4.1. Students will be able to communicate reasonably in different settings according to target audience tasks and situations |
| BLO4.2. Students will be able to convey their ideas effectively through an oral presentation |
| BLO4.3. Students will be able to convey their ideas effectively in a written paper |

**Learning objectives for the Bachelor of Social Science**

*Programmes:*

*Economics and Data Analytics,*

*Economics and Politics*

|  |  |
| --- | --- |
| **Learning Goals** | **Learning Objectives** |
| Students will be critical thinkers | ELO1.1. Students will be able to understand core concepts and methods in the key economics disciplines |
| ELO1.2. Students will be able to identify underlying assumptions and logical consistency of causal statements |
| Students will have skills to employ economic thought for the common good | ELO2.1.Students will have a keen sense of ethical criteria for practical problem-solving |
| Students will be technology agile | ELO3.1. Students will demonstrate proficiency in common business software packages |
| ELO3.2. Students will be able to make decisions using appropriate IT tools |
| Students will be effective communicators | ELO4.1.Students will be able to communicate reasonably in different settings according to target audience tasks and situations |
| ELO4.2.Students will be able to convey their ideas effectively through an oral presentation |
| ELO4.3. Students will be able to convey their ideas effectively in a written paper |