

Syllabus

Course title and number	ISTE-121 Computational Problem Solving in the Information Domain II
Instructor	Alan Mutka
Term	Spring
ECTS points	6
Credit hours (L+S+E)	4 (4+0+2)
Prerequisite	ISTE-120 or GCIS-123 or equivalent course
E-mail	alan.mutka@croatia.rit.edu
Office	ROOM 24
Office hours	MON and WED 13:00-16:00 https://rit.zoom.us/j/456613794

Meeting time

Section	Monday	Wednesday	Friday
Section 800	8.30 am – 10:30 am, LAB 3	8.30 am – 10:30 am, LAB 3	11.30 am – 1:30 pm, LAB 3
Section 801	10.30 am – 12:30 am, LAB 3	10.30 am – 12:30 am, LAB 3	1:30 pm – 3:30 pm

Program learning outcomes supported by this course:

- Analyze user, software and hardware requirements in the selection, development, integration, evaluation, and administration of information systems
- Design a computer-based solution for solving problems using mathematical and information technology principles and practices
- Apply software design principles and patterns to design and implement software solutions
- Build a software solution using contemporary software development methodologies, frameworks, libraries and tools.
- Design user interface and user experience for software solutions based on contemporary principles and practices
- Apply methods and techniques to design, build and manage a computer network system using requisite hardware and software

- Apply methods and techniques to design, build and manage a computer network system using requisite hardware and software
- Apply IT project management techniques to achieve specific project objectives including scope, quality, time, and cost
- Make decisions in computing practices based on professional, legal, ethical, social, and security principles
- Collaborate effectively as a team member or leader in various IT projects

Course Learning Outcomes

A student will be able to:

- CLO1:** Implement interactive graphical user interfaces using the event model
- CLO2:** Build algorithms for processing binary and text-based data structures
- CLO3:** Create advanced processing methods using multiple threads
- CLO4:** Design programming solutions using the client-server paradigm
- CLO5:** Use common data structures in programming solutions
- CLO6:** Apply basic software development, team work, and project management principles and practices

Course description

This is the second course in the introductory programming sequence required for all students majoring in Information Technology. Topics include GUI interface development, file I/O, traditional programming data structures, programming utilities and reusability, introductory project design and management concepts and other concepts as time permits. Emphasis is placed on the development of problem-solving skills. Large programming assignments are required.

Course materials

Required reading:

- Horstmann, C., & Safari, an O'Reilly Media Company. (2009). Big java, 4th edition (1st ed.). Wiley.

Additional reading:

- Horstmann, C., & Safari, an O'Reilly Media Company. (2018). Core java volume I-- fundamentals, 11th edition (1st ed.). Pearson
- Horstmann C, Safari, an O'Reilly Media Company. Core Java, Vol. II-Advanced Features, 12th Edition. 1st ed. Pearson; 2022.
- Horstmann, C., & Safari, an O'Reilly Media Company. (2011). Java for everyone: Compatible with java 5, 6, and 7, 2nd edition (1st ed.). Wiley.

Grading

Grading scale (minimum percent): A = 94; A- = 90; B+ = 87; B = 83; B- = 80; C+ = 77; C = 73; C- = 70; D = 60

F = <60 or < 75% over-all exam average

Students are required to have at least a 75% average of the exams in order to pass the course.

Components of evaluation:

Component	Points/%
Practical 1	8
Practical 2	15
Practical 3	10
QUI	6
Laboratory	19
Homework	17
Project	25
Total:	100

Course learning outcomes based grading table:

	P1	P2	P3	QUI	LAB	HW	PR	ECTS	Points
ECTS	0.48	0.9	0.6	0.36	1.14	102	1.5		
Points	8	15	10	6	19	17	25		
CLO1				1	4	2	3	0.6	10
CLO2	5			1	3	2	1	0.72	12
CLO3	3			1	4	8	6	1.32	22
CLO4		15		1	2	5	9	1.92	32
CLO5			10	1	6		2	1.14	19
CLO6				1			4	0.1	5

Attendance and participation:

- Each class presents new information that builds on the previous day's work. Missing a class can put the student's success in jeopardy. Attendance is mandatory and will be recorded.

Course schedule

Some Topics/Activities and assignments due may be changed.

Week/Lecture	Topic/Activity	Assignments due
W1 L1 W1 L2	Introduction GUI Layout manager Lab01 - GUIs	HW01 out
W2 L3 W2 L4	Events TextAreas, Menus, Inner Classes Lab02 – TextAreas, Menus, Event Handling	LAB01 due HW01 due, HW02 out
W3 L5 W3 L6	Binary Stream IO Binary IO Exercise LAB03 – GUIs and Inner Classes	Lab02 due HW02 due, HW03 out
W4 L7 W4 L8	Threads Intro Thread Control 1 Lab04 – Threads and Progress Bars	Lab03 due HW03 due, HW04 out
W5 L9 W5 L10	Thread Synchronization Thread Control 2 & Deadlock Lab05 – Threads and Binary IO	Lab03 due HW04 due, HW05 due
W6 L11 W6 L12	Practical Exercise: ByteIO, Threads	
W7 L13 W7 L14	Networking, Client and Server Multithreaded TCP Server and UDP Lab6/HW07 – Networking Client/Server	LAB05 due HW05 due, HW06 out
W8 L15 W8 L16	Lab06/HW07 Continued UTILITY TIME – For Instructor to use as needed Lab07 / HW08 – Object IO over Sockets	HW06 due
W9 L17 W9 L18	Lab07 / HW08 Continued Generics Sorting and Searching	Final Project Launch
W10 L19 W10 L20	Sorting More Complex Data Lab08 – Sorting Project Day – Communicatio, Protocol Design	Lab06 / HW07 due
W11 L21 W11 L22	Recursion Lab09 – Recursion and Directory Tree Walking	Lab07 / HW07 due

	Practical 2	
W12 L23 W12 L24	Project Day – Design Review Stacks, Queues and List Sets and Maps	Lab08 due
W13 L25 W13 L26	Lab10 – Data Structures Jars and Packages Project Day – Code Review	
W14 L27 W14 L28	Project Day - Interoperability Testing Project Day - Interoperability Testing Practical 3	Lab10 due
W15	Project Day - Interoperability Testing Makeups	

Policies

Course policies

Late submission (LAB and HW< assignments)

Late submission of lab and homework assignments will be penalized as follows:

- **Up to 1 days late: max grade possible is 90%**
- **Up to 2 days late: max grade possible is 80%**
- **More than 2 days late: no credit – no exceptions**

Technology in the classroom

The use of cell phones, tablets, laptops, and other personal electronic devices in class is not allowed, unless the instructor invites students to use them for a learning activity related to the course. These should be switched off and put away at the beginning of class. Students using such devices in class will lose points – these will be deducted at the end of the semester from the total number of points.

Written Assignments – Minimum Expectations

The written assignment needs to:

- be in line with the guidelines provided by the instructor

- be proofread (no spelling and grammar mistakes)
- be in Standard American English
- be organized in paragraphs
- use a proper citing and referencing style required by the instructor (e.g. APA, MLA, etc.)
- use proper formatting required by the instructor (e.g. font type and size, margins, spacing, alignment, etc.)

The instructor may refuse to accept the written assignment or deduct points if these expectations are not met.

Go to <https://bit.ly/citeref> to access guides on using different citing and referencing styles. Visit the RIT Croatia Writing Lab for any help you may need with writing.

General policies

It is the student's duty to thoroughly familiarize themselves with all the policies applied in the course. The policies can be found on the links below:

Equal Access and Disability Accommodation Policy

This policy outlines RIT Croatia's values regarding diversity, equity, and inclusion. [More](#)

Academic Honesty Policy

This policy provides information about academic honesty, academic dishonesty, and related procedures. [More](#)

Resources

Writing Lab

The Writing Lab is a free service which enables students to have a writing assignment for **any class** critiqued with suggested corrections and improvements by an instructor. Using this service improves the quality of the work the student hands in for a grade. [More](#)

AssessME

All practical assignments and exercises in the ISTE121 course, including but not limited to Examinations, Project, Laboratory, and Homework assignments, require the automated proctoring and learning validation software that uses advanced machine learning identity-verifying technology to ensure academic integrity and performance. Please read "**Getting Started with AssessMe**" for more detailed information about the software installation and usage.

Please be aware that:

- The student, their computer, and physical test-taking environment may be video recorded
- Students' activities within Visual Studio Code (code changes, keystrokes, clipboard) may be recorded.
- For the online assessment, the student may be asked to show identification to the camera.
- The student will require a quiet place to take the online assessments for the benefit of the students' concentration, and as interruptions (voices, another visible person on camera) may be flagged for potential academic dishonesty.
- All acquired data may be used to **prove unethical behavior in the Academic Honesty Process.**
- All data acquired during the assessment is encrypted, stored, and finally deleted under the *Exam Data Storage Policy*.

The requirements for Laboratory, Homework, Project and Practical Assignments are:

- Visual Studio Code with *AssessME.VSCODE plugin*
- Please submit the whole VSCODE project, **including the .assess directory**
- **The submitted source file without the corresponding .assess file will result in a grade of zero**

The requirements for online practical examinations are:

- Laptop or desktop computer with a fully-functional webcam and microphone
- Visual Studio Code with *AssessME.VSCODE plugin*
- A fully-functional webcam and microphone
- *AssessME Capture Essential* (the connection to the ZOOM is required)
- Please submit the whole VSCODE project, including the **.assess** directory
- **The submitted source file without the corresponding .assess file will result in a grade of zero**

Rules for reusing the code from other sources:

- When you copy code from an external source, whether a snippet of code or an entire module, you need to cite a source in the form of an inline comment in the code
- An inline comment should consist of the URL and the date of retrieval.
- If you adapt the code, you should also indicate "Adapted from" or "Based on" so that it is clear that the code is modified.