

---

# CSC 101

# THE SCIENCE OF COMPUTING I

---

## COURSE SYLLABUS

---

### BASIC INFORMATION

---

Course title: The Science of Computing I

Course number: CSC 101

Number of credits: 4

Term and year: Fall 2024

Instructors:

Dr. Lori Jacques ([ljacques@ut.edu](mailto:ljacques@ut.edu)), TECH 673F

Dr. Matt Lepinski ([mlepinski@ut.edu](mailto:mlepinski@ut.edu)), TECH 674F

Office hours:

Dr. Lori: Wed 12:30 - 3, Thurs 8:30 – 11, and by appointment

Dr. Matt: Mon 2:30-3:30, Tues 10:00 - 12:30, Fri 2:30-4:00, and by appointment

*Because we are coteaching this course, you may visit either professor's office hours for extra help. If you have a personal concern, such as about your grade, please see your assigned instructor.*

---

### COURSE TIMES AND LOCATION

---

Monday, Wednesday, Friday

Time: 10:00 – 11:10

Room: TECH 693 & 695

---

### COURSE DESCRIPTION

---

An introduction to computing. Topics include problem solving, algorithm analysis and development, computer programming in Python, data structures, and the object-oriented paradigm.

# COURSE LEARNING OUTCOMES

---

Upon successful completion of the course, each student will:

- (1) Be able to identify a problem's variables, constraints, and objectives;
- (2) Be able to represent algorithms in various ways (e.g., flowcharts, pseudocode);
- (3) Be able to write simple object- and non-object-oriented programs in a general purpose programming language (e.g., Python);
- (4) Have a basic understanding of common algorithms (e.g., searching and sorting) and their complexity;
- (5) Have a basic understanding of introductory data structures (e.g., arrays);
- (6) Have a basic understanding of the object-oriented paradigm; and
- (7) Have an understanding of computing as it applies in and how it affects the global context.

## TEXT AND MATERIALS

---

**Textbook:** Jean Gourd, Ankunda Kiremire, and Lorraine Jacques. 2023. The Science of Computing I. Provided to students in PDF format.

**Required Technology:** You will need a computer in class that you can write programs on, with Python and Thonny installed on it. You will also need access to a computer for writing papers, doing activities, etc.

## ASSESSMENTS & GRADING

---

Your grade will consist of the following graded items and weights:

Assessment	Weight
Exams (4)	40%
Programs (10)	37%
Room Adventure	10%
Paper (1)	5%
CT activities	4%
Attendance	4%

Grading will be based on the following scale:

Grade	Percentages
A=	92% to 100%
AB=	89% to 91%
B=	82% to 88%
BC=	79% to 81%
C=	72% to 78%
CD=	68% to 71%
D=	60% to 67%
F=	59% and below

---

## INSTRUCTOR'S AND/OR DEPARTMENT'S POLICIES

---

Attendance is part of your grade. Excused absences, as defined by the university, do not count against your grade. Both excused and unexcused absences will be allowed to make up class work, however only excused absences will be allowed to make up exams. And no, there will not be a Zoom option for those who cannot make it to class.

All work is expected to be submitted on time. Late submissions will be accepted but will be subject to a **5% deduction for each calendar day** it is late. No late work will be accepted after **December 1st**.

If you are having difficulties managing the work in this course, even if the reason has nothing to do with the course, it is expected that you will contact me ASAP so we can figure out what to do about it.

---

## FINAL EXAM

---

**Wed., December 9<sup>th</sup>, 11 - 1**

The final exam schedule for all courses can be found at: <https://www.ut.edu/about-ut/university-services/office-of-the-registrar/fall-final-exam-schedule>

---

## COURSE POLICIES

---

All UT policies applicable to this course can be found at <https://www.ut.edu/academics/center-for-teaching-and-learning/required-syllabus-disclosures>.

---

### A SPECIAL NOTE ON PLAGIARISM FOR THIS COURSE

---

In reality, software professionals seek help from peers, find code online, and do what it takes to get the project done. For this course, I expect you to do the same, **however**, I also expect you to cite your source(s) as comments in your code.

Examples:

```
# <classmate name> helped me with this section  
# Found this code at <website url>
```

**Failure to do so is plagiarism and will result in an F for the assignment.**

---

## CHATGPT AND OTHER LLM CODE GENERATING TOOLS

---

Large language models such as ChatGPT have taken the programming world by storm. Since the release of ChatGPT in 2022, each month has seen a high number of programmers using it and similar tools to generate, test, and fix code. Knowing how to use these tools effectively may be as important as knowing how to use a debugger well.

With that in mind, we are not banning the use of ChatGPT and other LLMs in this course. In fact, there will be times when we specifically ask you to use them and/or to analyze code we have generated with these tools. Most of the assignments for this course, however, cannot be solved by ChatGPT (we checked), so you should begin each programming assignment on your own, applying your own learning, and only use these tools as you would any other web-based resource:

- Ask how to do only a specific part of your code (e.g., how to read a file).
- Modify what it produces to fit the code you have already written.
- Check and double-check that the code meets the assignment's requirements.
- **Cite your use as you would any other source.**

## TENTATIVE COURSE SCHEDULE

---

The following is subject to change as I get to know you and your learning needs better.

Day	Topic	Book chapter(s)	Assignments
<b>Aug 26</b>	Intro to the Course	0	Paper 1 assigned
<b>28</b>	The Science of Computing	1 and 2	
<b>30</b>	CT: Intro & Decomposition	3	
<b>Sept 2</b>	<b>NO CLASS (Labor Day)</b>		
<b>4</b>	Intro to Python	4	
<b>6</b>	Variables and Operators	5	
<b>9</b>	Control statements	6	Program 1 assigned; <b>Paper 1 DUE</b>
<b>11</b>	<a href="#">in-class practice</a>		
<b>13</b>	Subroutines	7	
<b>16</b>	CT: Algorithms	8	<b>Program 1 DUE</b> ; Program 2 assigned
<b>18</b>	Representing Algorithms	8	
<b>20</b>	<a href="#">in-class practice</a>		
<b>23</b>	<b>EXAM 1</b>		<b>Program 2 DUE</b> ; Program 3 assigned
<b>25</b>	Arrays	9	

27	For loops with Arrays	9	
30	File I/O and Split	Focus 2	<b>Program 3 DUE</b> ; Program 4 assigned
Oct 2	<a href="#">in-class practice</a>		
4	Strings	10	
7	<a href="#">in-class practice</a>		<b>Program 4 DUE</b> ; Program 5 assigned
9	<b>EXAM 2</b>		
11	CT: Abstraction	11	
14	Searching (concept)	12	<b>Program 5 DUE</b> ; Program 6 assigned
16	Sorting (concept)	12	
18	Searching & Sorting with Arrays	12	
21	Searching & Sorting with Arrays (cont.)	12	<b>Program 6 DUE</b> ; Program 7 assigned
23	Scope of variables	13	
25	Recursion	13	
28	<a href="#">in-class practice</a>		<b>Program 7 DUE</b> ; program 8 assigned
30	<b>EXAM 3</b>		
Nov 1	Intro to OOP	14	
4	Coding Classes/Objects	14	<b>Program 8 DUE</b> ; Program 9 assigned
6	Using Objects	14	
8	<a href="#">in-class practice</a>		
11	Room Adventure Activity	15	<b>Program 9 DUE</b> ; Program 10 assigned
13	Room Adventure Activity	15	
15	Inheritance	17	
18	<a href="#">in-class practice</a>		<b>Program 10 DUE</b>
20	Overloading	17	
22	<a href="#">in-class practice</a>		
25	CT: Debugging	16	
27&29	<b>NO CLASS – TDay Break</b>		
Dec 2	Abstract classes	17	
4	<i>Extra help / catch-up</i>		
6	<i>Extra help / catch-up</i>		<b>Room Adventure DUE</b>
Dec 9	<b>EXAM 4 (Final)</b>		<b>11:00 – 1:00</b>