CSE 007: Introduction to Programming

Spring 2022

**Instructor Information:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Email** | **Office** | **Office Hours\*** |
| Prof. Sharon Kalafut | smk5@lehigh.edu | PL 200A | MW 3-4 pm  TR 11 - 12 pm (Virtual) |
| Prof. Kallie Ziltz | krz216@lehigh.edu | PL 200B | T 2-4pm  R 9:30 - 11 a (Virtual) |

\*Regularly scheduled office hours are first come, first served (no appointments req’d); both instructors are also available by appointment (schedules permitting). Zoom links for virtual office hours can be found on CourseSite.

**Course Description**: *Problem-solving using the Java programming language. Data types, control flow, methods, arrays and multidimensional arrays, objects, inheritance, polymorphism, and breadth of computing. Includes laboratory.*

**Prerequisites:** None. No prior programming experience is required.

**Format of Course:** This course is scheduled to be completely in-person after the first week of classes, but incorporates both synchronous and asynchronous work. You will be asked to complete a series of tasks and activities before class (at your own pace) and during class.

* **Mondays** & **Wednesdays**: We will meet **synchronously** for a combination of interactive lecture and hands-on work. *You should always come to class prepared, and ready to be active and attentive.*
* **Fridays**: We will meet **synchronously** to complete a 50 minute lab session.

**Required Text:** Our course text is an online book: *CSE7: Introduction to Programming*, published by zyBooks. To access the textbook:

1. Sign in or create an account at [learn.zybooks.com](http://learn.zybooks.com)

2. Enter zyBook code: **LEHIGHCSE007Spring2022**

3. Subscribe ($77)

**CourseSite:** You are responsible for checking the class CourseSite page daily. Class lectures, labs, homework assignments, and other supplemental materials will be posted frequently.

**Piazza:** Piazza is a Q&A platform that supports discussion and collaborative learning. During the first class, you will receive instructions for joining, as well as expectations to keep in mind when posting.

**Platform:** All programs will be submitted as Java source files; thus, no specific IDE (Integrated Development Environment) is required.

**Student Learning Objectives:** By the end of this course, students should be able to:

1. Utilize computational thinking strategies to solve complex and authentic problems.
2. Communicate effectively their understanding of various topics broadly related to computer science, computer systems and computer applications.
3. Write Java code to implement fundamental programming statements, including assignment, comparison, selection, and iteration.
4. Implement and evaluate algorithms for searching and sorting arrays (including multidimensional arrays).
5. Incorporate methods in Java programs to modularize existing code and address various program requirements.
6. Develop an understanding of object-oriented programming, and implement into Java programs basic concepts including classes, inheritance and polymorphism.
7. Compile, run and test Java code to deliver fully working programs, which requires both of the following:
   1. Employ basic debugging techniques to address common errors and exceptions.
   2. Use appropriate documentation practices to describe programs and complex code.

**Grading:** Final course grades will be calculated using the following breakdown:

|  |  |
| --- | --- |
| Homework: | 20% |
| Labs: | 10% |
| Attendance, Prep & Participation Activities: | 15% |
| Midterm Exams (2): | 30% |
| Final Exam: | 25% |

Final course grades will be assigned based on your final grade average as follows:

92-100: A 82-87: B 72-77: C 62-67: D

90-91: A- 80-81: B- 70-71: C- 60-61: D-

88-89: B+ 78-79: C+ 68-69: D+ 0 - 59: F

Homework, lab, prep activity, and exam grades will be recorded on CourseSite. However, any average overall course grade displayed on this site may not be accurate and the student should refer to the above breakdown when calculating their current average.

**Grading Appeals:** Specific grading disputes (points totaled incorrectly, partial credit appeals, missing assignments on CourseSite, etc) should be brought to my attention via email *within one week* of return of the assignment. Contacting me within this time frame does not guarantee a grade change, only my consideration. Appeals outside of this timeframe will not be considered. After the final exam, no grades may be contested. Do not contact the graders with any disputes.

**Attendance**: Attendance during lectures and labs is mandatory. Attendance will count towards the participation grade category (described in detail below).

**Homework:** Homework assignments will be assigned weekly, except during exam weeks. Assignments may include conceptual questions with written responses or complete Java programs. All assignments will be submitted through CourseSite. The only acceptable file types will be .java (for programming assignments) and .pdf (for written responses). You may handwrite the written answers but submitted PDFs must be legible and readable. **Late homework will not be accepted, but your lowest homework grade will be dropped at the end of the semester.**

**Labs**: Labs will be held every Friday (except during exam weeks), available to be completed either in-person or virtually. Information about coming to campus for labs will be forthcoming. I encourage you to do your best to complete the lab synchronously while the graders and myself are available to answer questions. However, I understand you may sometimes require more time to finish an assignment. You will have until the beginning of the next lab (held on the following Friday) to submit your completed lab without a late penalty. **Late lab submissions will not be accepted, but your lowest lab grade will be dropped at the end of the semester.**

**Prep Activities:** Each week there will be specific reading activities from our online textbook. Although you will complete the readings through ZyBooks, there will be a quiz to complete each week to assess your understanding of the material. These quizzes will open on Tuesdays and close on Sundays at 11:59 pm. It is important to the flow of the class that you complete the readings ahead of Monday classes.

**Participation Activities:** Lectures will be interactive as much as possible, so it is likely that you will also complete activities during class that will be submitted on CourseSite. Although all of these activities will be assigned grades, know that they are all participation based. **Completion alone, regardless of correctness, will dictate your grade**, and all participation is due on Saturdays @ 11:59 pm (considered the end of the week). **No late assignments will be accepted.**

**Exams:** There will be two midterms and a final exam in this course. The exams will take place virtually and synchronously. The midterms will likely be held during our allotted class times on Fridays (using the full 75 minutes as scheduled by the Registrar). The tentative dates for the exams are as follows:

* Exam 1: Friday, February 25th, 2022
* Exam 2: Friday, April 8th, 2022
* Exam 3: TBD (scheduled by registrar)

**Makeup Policy:** In general, makeup exams and extensions for homework, labs, and prep activities will not be given. That said, **we are all human!** If something serious and beyond your control arises, contact your Instructor (via email) **before missing the work.** Cases will be handled on an individual basis.

**General Program Grading Policy:** While each programming assignment (including those found in HWs, Labs and Exams) will be graded using individualized rubrics, each follows generally the same breakdown (included below). In this way, we are able to reward significant partial credit for programs that might compile (and are thus syntactically correct), but don’t necessarily run as expected.

* Appropriate Programming Practices: 15%\*
* Compile: 25%
* Run/Output: 35%
* Meets requirements as listed: 25%

\*[More information about appropriate programming practices & standards available here](https://docs.google.com/document/d/1_MQ9xRK53zNNa1IzTpwnrns53BIH1oREQz7DhGlmU74/edit?usp=sharing)

**Accommodations for Students with Disabilities:** Lehigh University is committed to maintaining an equitable and inclusive community and welcomes students with disabilities into all of the University’s educational programs. In order to receive consideration for reasonable accommodations, a student with a disability must contact Disability Support Services (DSS), provide documentation, and participate in an interactive review process. If the documentation supports a request for reasonable accommodations, DSS will provide students with a Letter of Accommodations. Students who are approved for accommodations at Lehigh should share this letter and discuss their accommodations and learning needs with instructors as early in the semester as possible. For more information or to request services, please contact Disability Support Services in person in Williams Hall, Suite 301, via phone at 610-758-4152, via email at indss@lehigh.edu, or online at <https://studentaffairs.lehigh.edu/disabilities>.

**The Principles of Our Equitable Community:** Lehigh endorses The Principles of Our Equitable Community ([available here](https://diversityandinclusion.lehigh.edu/sites/diversityandinclusion.lehigh.edu/files/Principles_Five%20Languages_GateFold_Pages.pdf)). We expect each member of this class to acknowledge and practice these Principles. Respect for each other and for differing viewpoints is a vital component of an inclusive learning environment inside and outside the classroom.

* **Our Classroom Community:** At the beginning of the semester, everyone (professor, graders, students) will reaffirm the Principles. More info will be provided on the first day of class.

**Academic Integrity:** All programming assignments, labs, and tests submitted will be considered graded work and must be completed on an individual basis (unless stated otherwise). *All graded work is subject to be checked for plagiarism and/or software similarity.* Homework and labs may be *discussed* with other students as well as with the professor and graders. However, in coding or working on a particular assignment, each student works alone (without other students or other students’ code or papers) and certifies that what is submitted accurately represents the student’s own understanding of the material expressed in the student’s own words. Absolutely no consultation is permitted on tests. **Students may not consult partial or complete solutions of the problems that have been prepared by anyone else**. Students are encouraged to use resources such as zyBooks, class lectures and activities, and Java API Documentation ([from Oracle](https://docs.oracle.com/javase/8/docs/api/overview-summary.html)). All submitted work is subject to assessment of software similarity.

* While there are *constructive* and *appropriate* ways to collaborate, using anyone else’s solution or sending your work to someone else to be fixed are **neither**.
  + Additional examples of academic integrity violations/non-violations specific to CSE are [available here](https://docs.google.com/document/d/12tF2xaYzq9GcgsIMfjKXpTPBIHUGF-GFw53PhcGqwrM/edit?usp=sharing).
* Review the [Statement of Academic Integrity](https://studentaffairs.lehigh.edu/content/academic-integrity-resources) issued by the undergraduate student senate. Ignorance of rules as listed in the Code of Conduct does not excuse a student for violations of Academic Integrity.

**Tutor Policy:** For the purposes of this course, there are two types of tutors: approved tutors and unapproved tutors. Approved tutors include any tutor that works for Lehigh’s Center for Academic Success (intutor@lehigh.edu or 610-758-4154). All other tutors or sources of outside help are considered unapproved. Unapproved tutors may only help you understand reading material, study for tests and to do extra programming exercises that are unrelated to any graded assignments. They may not help you with any homework or graded programming assignment in any way. Approved tutors can help any way an unapproved tutor can, but can also help you with your assignments by reviewing complex concepts, general troubleshooting and debugging assistance, as well as discussing strategies for solving a problem with you. Even though the tutoring sessions offered through the Center for Academic Success are group-sessions, it should be clear that this is not a group with which you can complete your assignments. **No one at the session (tutor or student) should share any code** and if there are multiple students at a group tutoring session, **all troubleshooting assistance must be done on a one-on-one** basis; i.e., they cannot help multiple students with their assignments at the same time. You are responsible for ensuring that you and any tutor you work with understand and comply with these academic integrity and tutoring policy statements.

* If a tutor helps you with an assignment, you must include “with assistance from tutor name (tutor email address)” below your name in the header comment.

**A Note for Spring 2022:**

As indicated by our Provost in early January, we will begin our semester by meeting remotely for the first week. After this, as long as it remains safe, we will meet in our assigned classroom unless otherwise indicated by the instructor or University. In our physical classrooms (and other indoor spaces), Lehigh is currently following a policy requiring everyone to wear face coverings when in public spaces inside buildings on our campus and to maintain social distance. This policy applies to our physical classroom. Thank you in advance for following this rule. Students who do not wear a face covering during in-class meetings will be reminded to put their face covering on. If they do not do so, they will be asked once again to do so or leave the classroom.

//Object is one instance of a class, memory is allocated to an object

//Object has specific values for its attributes/data members, and methods called on the object to manipulate those attributes

//- for private, + for public, UML diagram

//Procdural data declared in main, direct access to data in main, obj data declared in methods and edit that way, access data through methods

Object

Objectreference.methodmember (must be public)

. is access operator

Accessor or getter doesn’t change variable

Constructors initialize data members, have the same name as class itself, no return type, no void, public

Invoked automatically when new is used

Noark constructor, giving values to variables to

Threeark constructor, allows user to initialize different variables as you create a new object of the class.

Deep copy, initialize a new object and initialize the class to .methods for each initialized object

Can return a newark instance of an object and can get memory location used in new

Scanner scnr = new Scanner(System.in);

Student s1 = initialize(scnr);

Public static Student initialize(Scanner scnr){

Ahusbdhuasdhoahsodasd

Return new Student(1,2,3);

}

Create object, and can see default by calling new Object and not assigning it to a variable

Class Scanner java.util.Scanner

Scanner scnr = new Scanner(System.in);

Creating object input with one parameter

Instance member tied to an object at that instance- private variables

Static member tied to class and shared between all class instances

Data or methods shared by all objects of a class

Static data members belongs to the class, static data members accessible by all objects of the class

Static method is called using the class name, not object

Instance methods can invoke everything

Static methods can only invoke static members

Immutable objects – objects cannot be modified (no mutator method)

All data members are private

No mutator methods in class

Class String is immutable

Scope of variables sit within each specific class

Local variables have scope inside their method or where they are declared (just like a for loop)

Local variable takes precedence over class variable if they have the same name

IMPORTANT IF THEY HAVE THE SAME NAME

This keyword used as reference

Refers to class instance object itself

This() invokes another constructor if used inside a constructure, should be first statement

This.name compilers knows it goes with class, not local variables

Can iterate an array of objects, associating memory locations to instances where each has a set of the default variables

Objects store data and manipulate data with methods and classes

Instantiating an object holding data from a class

Private variables and methods are help inside a class/object member

Static members known to all v instant member only known to that object

Private can only access variable from method from that class

Public can be accessed through the class itself

Use . and .get

When you instantiate, you call constructor that instantiates that object, defined to give data certain values

Inherit, from base class

Derived class

Class derivedClass extends baseClass{

}

Class circle inherits all instane variables and methods of shape

Shape and circle are called respectively

Base, derived

Parent child

Super sub super. Gets to the parent class in case two methods are same

Protected – derived class has access to private variables

Else then the derived class needs to look at the data through methods

Every class creates has a super class, class Object

Even user defined classes

Class w/ modifier final, class can not be inherited/extended

Overriding a method in derived class, base.

UML:

* = private

# = protected

+ = public

*Italicized* = abstract (definition no implementation, ex. Public abstract class shape{}, public abstract void perimeter{})

toString provided by abstract class providing toString,

abstract definitions get none, just returns it, no code, just

public abstract String stringy();