

**COURSE NUMBER:** CSCI125**CREDITS:** 3**COURSE TITLE:** INTRODUCTION TO COMPUTER SCIENCE
AND PROGRAMMING II (JAVA)**PREREQUISITES:** Math 12 or equivalent
and CSCI 120**CO-REQUISITES:** CSCI127**Weekly Hours:** 3**Lecture:** 3**Lab:** -**Total Hours:** 3**Total Weeks:** 13

COURSE DESCRIPTION: This course is a thorough introduction to computer science and computer programming, suitable for students with some computer science background. It is designed for students who will major in computing science or a related program. Students will learn fundamental concepts and terminology of computer programming and acquire basic programming skills in the Java programming language. Topics covered are: Primitive and abstract data types; elementary data structures; fundamental algorithms; algorithm analysis; basic object-oriented programming and software design; specification and program correctness; and historical aspects of computing science.

TEXTBOOK: Tony Gaddis, Godfrey Muganda: Starting Out with Java: From Control Structures through Data Structures, 3rd Edition. Pearson. 2016. ISBN 0134038177 • 9780134038179.

LEARNING OUTCOMES:

Upon successful completion of this course, students should be able to:

- Cite some historical facts of computer science and programming.
- Demonstrate how to choose primitive data types appropriately.
- Know the difference between primitive data types and reference data types.
- Describe and apply techniques to document programs.
- Demonstrate how to use control structures appropriately.
- Demonstrate how to use text files for input and output.
- Know how to construct safe programs.
- Know how to apply the concept of functional decomposition.
- Explain and use simple abstract data types such as list, stack and queue.
- Demonstrate how to design, and use classes.
- Demonstrate how to use modeling tools such as UML, for object oriented design.
- Know data structures from the Java Collections API such as arrays, lists and dictionaries.
- Demonstrate how to use single and multi-dimensional arrays.
- Know and analyze fundamental searching and sorting algorithms.
- Explain and apply the concepts of Object Oriented Programming.
- Define and use recursive algorithms for problem solving.



COURSE CONTENT:

Week	Topic
Week 1	Introduction to Java, OOP, and Program Design
Week 2	Primitive data types and their operations
Week 3	Decision and loop control structures
Week 4	Introduction to File Input and Output / Midterm Exam 1
Week 5	Methods
Week 6	Introduction to Classes and Objects, UML diagrams
Week 7	Introduction to Arrays and the ArrayList Class
Week 8	Basic Sorting Algorithms
Week 9	Searching Algorithms
Week 10	Recursion / Midterm Exam 2
Week 11	Abstract Data Types
Week 12	Inheritance and Polymorphism
Week 13	Exception Handling

EVALUATION:

Classwork / Pop-Quizzes/ Participation	15%
Midterm Exam 1	25%
Midterm Exam 2	25%
Final Exam	35%
Total	100%



CHEATING:

Students cheating on tests and exams will receive a “F” grade in this course.

If a student misses an exam, a mark of zero will be assigned unless there are extenuating circumstances. In such cases, the proportion of grade assigned to the missed exam will be added to the proportion assigned to the final exam. The final exam will be held during exam week. NO consideration will be given to any student wishing to write the exam at any other time than that assigned.

It is a student’s responsibility to know and follow the school’s policies regarding cheating on exams.

The school’s policy regarding electronic devices is that any student who has a cell phone or other unauthorized electronic device (i.e. Ipad, laptop, playbook, etc.) on their person or around their desk during an exam will be guilty of cheating and will a grade of “F” for the course.

ATTENDANCE:

If a student is absent for more than 50% of the course they will not be allowed to write the final exam.