Course Number, Course Title and Semester Hours

CSC 130, Computer Science I, 4 sh.

Course Coordinator

Shannon Duvall

Current Catalog Description

This introduction to programming and problem solving emphasizes applications from quantitative disciplines and incorporates weekly group lab experiences. Prerequisites: MTH 100 or its exemption. Offered fall and spring.

Textbook

A Comprehensive Introduction to Object-Oriented Programming with Java, by C. Thomas Wu

References

None.

Course Goals

- To work in teams, as both leader and participant
- To have the knowledge and skills needed to remain current in the field and engage in life-long learning
- To solve problems in an object-oriented paradigm
- To increase the students' technical competence and confidence

Prerequisites by Topic

- MTH 100: Algebra skills

Major Topics Covered in the Course

- PF1: Fundamental programming constructs (7)
- PF2: Algorithms and problem-solving (5)
- PF3: Fundamental data structures (3)
- PL4: Declarations and type (1)
- PL6: Object-oriented programming (2)
- SP2: Social context of computing (2)
- SE2: Using APIs (1)
- SE3: Software tools and environments (1)

Learning Objective/Outcomes with Cross Reference to CSC Program Outcomes

- Understand the basics of problem-solving and creating pseudocode (Outcomes 5,8)
- Know how to use an API (Outcome 8)

- Know the basics of Java and object-oriented programming (Outcomes 5, 9)
- Learn the ACM Code of ethics and be able to make decisions based on this and other prevalent ethical codes (Outcome 4)
- Learn to work on programming projects in groups (Outcome 2)

Laboratory projects (specify number of weeks on each)

There is approximately one project per week. Most are completed individually, but several will be completed using pair programming.

- Declaring, creating, and using objects
- Calculating numerical data
- Using conditional statements
- Using loops
- String manipulation
- Defining classes that work together
- Using arrays
- Using file I/O and exception handling

Estimate Curriculum Category Content in semester hours. Maximum number of semester hours per course is 4. Times should be in increments of .5. (Advanced is a topic requiring previous core materials – most likely found in a 300 or 400 level class.)

Category	Core	Advanced
Data Structures	.5	
Algorithms	2	
Software Design	.5	
Computer Architecture		
Programming Languages	1	

Oral and Written Communications

Every student is required to submit at least __0___ written reports (not including exams, tests, quizzes, or commented programs) of typically ___0__ pages and to make __0___ oral presentations of typically __0___ minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues

Students will learn the major topics of computer ethics. They study three major decision-making theories as well as the ACM code of ethics and the Ten Commandments of Computer Ethics. They are asked to apply their knowledge to case

studies. (1.5 hours) They are assessed on this material in the final exam and the second in-class exam.

Theoretical Content

- APIs and IDEs (2 hours)
- Computer Ethics (2 hours)

Problem Analysis

Each homework assignment and in-class exercise is an opportunity for students to get a problem and create a solution. Problems included games, photo manipulation, art creation, and AI. For each problem the student must decide how best to implement an object-oriented solution.

Solution Design

Students learn the basics of programming and object-oriented solution design. They sometimes begin with a UML diagram or pseudocode for a solution. For each problem they design, implement, and test a solution in Java.

Course Assessment (List date of pre or post assessment)

Proposed changes from last offering (<u>Pre assessment</u>)

This semester is the first time I have taught in the traditional three-day-a-week format without a specific lab time. Therefore, the class will have more of the small-scale assignments with the larger assignments given for homework. Another consequence of the new format will be fewer assignments completed in groups. Since the class will be taught in a lab, much more class time will be spent in hands-on programming practice.

Learning outcomes to be assessed, how they will be assessed and success criteria (<u>*Pre assessment*</u>)

Item 9: Students will be able to solve problems using procedural, functional, and object-oriented programming paradigms.

All students will be learning to program in an object-oriented paradigm only. Since each assignment, exam, and daily assessment will require knowledge of object-oriented problem solving, the best measure of success is the overall measure of success in the course. Success will be defined as an average of 70% or greater for overall course grades.

Assessment data and analysis (Post assessment)

The following data show the students problem-solving ability using object-oriented programming. Two sections of CSC 130 are included, for a total of 47 students.

Overall grade:

A	B	C	D	F
40%	30%	23%	6%	0%

This overall grade shows 93% meet the criteria of 70% or higher on the overall grade. Since exams and quizzes are the only times when students solve problems on their own without other classmates or tutors, I will also include that data:

Quiz, exam, and final exam average:

A	B	C	D	F
17%	28%	17%	23%	15%

By this criteria, 62% of the students meet the criteria of 70% or higher. ***Overall, I believe that the course successfully achieved the stated outcome.*** However, I will place more emphasis on individual achievement in the future to be sure that all students are meeting the given learning objectives throughout the course.

Proposed changes for next offering (Post assessment)

For future offerings of CSC 130 it is important for me to learn effective classroom management for a lab environment. I would like to incorporate more in-class programming more effectively. This semester the in-class programs did not necessarily get graded and therefore the students did not always take the assignments seriously, and some relied on others to solve the problems for them. Also, it is difficult to ensure that students aren't working on computers during lecture or using inappropriate applications during class. Lab management software would help with this issue.

Group work seems to be an issue in general, as several students are unable to complete quizzes on their own because they are not learning from the group homework and class work assignments. In the future I will see if giving more individual homework assignments helps with this issue.