General Education Course Inclusion Proposal

Quantitative Reasoning and Analysis

This proposal form is intended for departments proposing a course for inclusion in the Northern Michigan University General Education Program. Courses in a component satisfy both the Critical Thinking and the component learning outcomes. Departments should complete this form and submit it electronically through the General Education SHARE site.

Course Name and Number: CS 120 - Computer Science I

Home Department: Mathematics and Computer Science

Department Chair Name and Contact Information (phone, email): J.D. Phillips (227-2020,jophilli@nmu.edu)

Expected frequency of Offering of the course (e.g. every semester, every fall): every semester

Official Course Status: Has this course been approved by CUP and Senate? YES

Courses that have not yet been approved by CUP must be submitted to CUP prior to review by GEC. Note that GEC is able to review courses that are in the process of approval; however, inclusion in the General Education Program is dependent upon Senate and Academic Affairs approval of the course into the overall curriculum.

Overview of course (please attach a current syllabus as well): *Please limit the overview to two pages (not including the syllabus)*

A. Overview of the course content

This course covers introductory Java, learning it to the point of mastering conditionals, loops, classes, and methods. An introduction to Object-Oriented Design is also given. At the conclusion of the course, the student should be able to program well enough to solve certain mathematical problems and to draw and animate basic images in a graphics window using the Java programming language.

B. Explain why this course satisfies the Component specified and significantly addresses both learning outcomes

Critical Thinking:

Evidence: In writing programs, students have to decide which syntax and strategies that may have been learned in class are appropriate for writing a particular program. This is not obvious and requires significant consideration of alternatives.

Integrate: All programming requires this. A student must be able to develop a strategy to solve a certain problem. Each problem is unique and must be solved by applying fundamental skills in a novel setting. Problem solving techniques are covered in class, but on exams and on programming assignments, students must be able to integrate their own analysis of the given problem with the techniques given in class.

Evaluate: Students have to be able to tell that their program produces correct output under a possibly infinite set of different input data. In part, this can be done by running the program sample input data, but it is also critical that students develop a mental model of the computation, evaluate the computer symbolically in their mind, and convince

themselves that the program produces correct output on every possible input. This skill is VERY difficult to develop and is a significant differentiator between strong and weak programmers.

Quantitative Reasoning & Analysis:

Calculation: Mathematics is present throughout the programming process. In a very real sense, programming is the act of developing a mathematical and logical model whose operation transforms the available input data into a desired output. For example, if a program is to create an amortization table for loans, the program must take the loan amount, interest rate, and payment schedule to compute the amortization table. If the assignment is to draw graphics, the programmer must convert the mental image of the graphics into Cartesian coordinates, typically using algebra.

Analysis/Application: In fact, this is the essence of computer programming. All programs and all exams will require the student to be able to derive mathematical information from a given set of input data.

Interpretation: A computer program is a description of information presented in a mathematical form. On all exams, students will be required to interpret the meaning and/or analyze the results of prewritten programming code presented to them.

C. Describe the target audience (level, student groups, etc.)

The students in this class are largely freshmen. They are not expected to have programmed a computer before.

D. Give information on other roles this course may serve (e.g. University Requirement, required for a major(s), etc.)

It is presently required in for the Computer Science, Mobile & Web App Development, and Mathematics majors and is an elective in the Business Computer Information Systems major.

E. Provide any other information that may be relevant to the review of the course by GEC

PLAN FOR LEARNING OUTCOMES
CRITICAL THINKING

Attainment of the CRITICAL THINKING Learning Outcome is required for courses in this component. There are several dimensions to this learning outcome. Please complete the following Plan for Assessment with information regarding course assignments (type, frequency, importance) that will be used by the department to assess the attainment of students in each of the dimensions of the learning outcome. Type refers to the types of assignments used for assessment such as written work, presentations, etc. Frequency refers to the number of assignments included such as a single paper or multiple papers. Importance refers to the relative emphasis or weight of the assignment to the entire course. For each dimension, please specify the expected success rate for students completing the course that meet the proficiency level and explain your reasoning. Please refer to the Critical Thinking Rubric for more information on student performance/proficiency in this area. Note that courses are expected to meaningfully address all dimensions of the learning outcome.

DIMENSION	WHAT IS BEING ASSESSED	PLAN FOR ASSESSMENT
Evidence	Assesses quality of information that may be integrated into an argument	Task Type: Programming Assignments: Every program will require students to choose the material they have learned in class that they believe will be relevant to completing the program. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to choose the material they have learned in class that they believe will be relevant to designing and writing a computer program. Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade Expected Proficiency: 75%
Integrate	Integrates insight and or reasoning with existing understanding to reach informed conclusions and/or understanding	Task Type: Programming Assignments: Every program will require students to integrate their analysis of the problem with the programming strategies they have learned in class to create a functional program. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to integrate their analysis of the problem with the programming strategies they have learned in class to create a functional program (on paper). Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade Expected Proficiency: 75%

Evaluate	Evaluates information, ideas, and activities according to established principles and guidelines	Task Type: Programming Assignments: Every program will require students to test their program to verify that the program works properly. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to carefully examine their own code and to model it on paper to demonstrate that their program would work if it had been typed in. Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade Expected Proficiency: 75%

PLAN FOR LEARNING OUTCOMES
QUANTITATIVE REASONING AND ANALYSIS

Attainment of the QUANTITATIVE REASONING AND ANALYSIS Learning Outcome is required for courses in this component. There are several dimensions to this learning outcome. Please complete the following Plan for Assessment with information regarding course assignments (type, frequency, importance) that will be used by the department to assess the attainment of students in each of the dimensions of the learning outcome. Type refers to the types of assignments used for assessment such as written work, presentations, etc. Frequency refers to the number of assignments included such as a single paper or multiple papers. Importance refers to the relative emphasis or weight of the assignment to the entire course. For each dimension, please specify the expected success rate for students completing the course that meet the proficiency level and explain your reasoning. Please refer to the Rubric for more information on student performance/proficiency in this learning outcome. Note that courses are expected to meaningfully address all dimensions of the learning outcome.

DIMENSION	WHAT IS BEING ASSESSED	PLAN FOR ASSESSMENT
Calculation	Ability to perform mathematical/numerical operations.	Task Type: Programming Assignments: Every program will require students to program some form of mathematical computation, either to solve a mathematical problem or to lay out images on a graphics window. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to solve a mathematical problem or to lay out images on a graphics window Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade Expected Proficiency: 75%
Analysis/Application	Ability to manipulate quantitative data to produce new data. Ability to use data to make judgments and draw conclusions.	Task Type: Programming Assignments: Every program will require students to program some sort of mathematical formula to produce an answer. About half of these programs will be strictly mathematical in nature. About half will use the data transformed using linear algebra to construct images in a graphics window. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to program some sort of mathematical formula to produce an answer. About half of these programs will be strictly mathematical in nature. About half will use the data derived from formulas to construct images in a graphics window. Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade

		Expected Proficiency: 75%
Interpretation	Ability to explain information presented in mathematical forms (e.g. equations, graphs, diagrams, tables, and words)	Task Type: Programming Assignments: Every program will have some mathematical component. A program is itself a mechanism for interpreting data and expressing it in different forms (for example, in a graphics window). Students are also expected to comment their code in natural language to explain how this information is being processed. Frequency: Approximately six times per semester Importance: Approximately 50% of the grade Expected Proficiency: 75% Task Type: Quizzes & Exams: Every examination and at least 80% of all quizzes will require students to program some mathematical component. The quizzes that don't require programming will require students to analyze and explain in natural language how certain information is to be processed correctly. Frequency: Approximately ten times per semester. Importance: Approximately 50% of the grade Expected Proficiency: 75%

CS 120 section 1, Winter 2015

Instructor: Michael Kowalczyk
Office: 2222 Jamrich Hall
Office Phone: 227-1600
Office Hours: 11:00am –11:50am Mon/Wed/Thurs/Fri or by appointment
Email: mkowalcz@nmu.edu
Class Meetings: 12:00noon – 12:50pm Mon/Wed/Thurs/Fri in 3309 Jamrich Hall
Course Website: https://educat.nmu.edu

Overview:

This course is an introduction to writing computer programs using Java. Although I assume you have never used a programming language before, you will probably find the course challenging and interesting even if you have.

Prerequisites:

Mathematics Placement recommendation of MA100 or higher, or CS101 or CIS110.

Textbook:

None required, but I have listed some useful E-books and other resources in the course website.

Equipment:

You will need a laptop computer with a web browser and Internet access. You will also need to do some software installs (see "Software installations and your first program" and "Homework 0" in Educat for full details). If for some reason you don't have a laptop computer, talk to me as soon as you can, since we will be using them for in-class exercises.

Grading:

Grades will be based mainly on exams, but also on homework assignments, labs, and quizzes. Homework assignments are weighted based on their size and complexity.

10% Labs and homework
10% Quizzes
40% Midterm exam
40% Final exam

Handing in Programs and Late Policy:

Some work (programming assignments) is handed in electronically while other work (labs, worksheets, quizzes, and exams) is submitted in person. I expect that you aim to hand in programming assignments 3 days before the deadline (you may still revise it as often as you like), as deadlines are strictly enforced (one minute late is still late), and late work gets no credit. It is your responsibility to pace yourself accordingly. If for some reason you are having trouble handing something in, you can email it to me as an attachment ***before*** the deadline.

Exam Dates & Schedule Conflicts:

The midterm and final exams are administered *on paper only*; no book, no computer, no notes. The midterm exam will be during our regular class meeting on Friday, February 27. The final exam will be on Monday, April 27 from 12:00noon until 1:50pm. Any conflicts with the exams (due to religious observances, other coursework, intercollegiate athletics, etc) must be made known to me within the first two weeks of the semester.

Laptop Use:

Some class meetings will be labs, in which your laptop is required. On other days, we will have lecture or discussion and I will need your complete attention (laptops closed).

You are responsible for keeping your laptop in good working condition and making frequent backups of your work. Note that the helpdesk does not backup your work if they need to fix your laptop (unless you want to pay them a fee), so make frequent backups to hardware external to your laptop *before* a crisis strikes.

Academic Conduct:

I work hard, with honesty and integrity; I expect my students to do the same. Every assignment must be written entirely by you. There are precisely two instances where including program source code from elsewhere is acceptable:

- You may include any code that I give out in my lab tutorials and lecture notes, without citation.
- Any other code that you didn't author **must** be accompanied with a full citation (this includes people, websites, books, etc.). Indicate clearly which lines of code you didn't write, and where it came from.

The best way to help others succeed in the course is by explaining concepts and working through examples – not by "sharing" source code.

Course objectives:

CS 120 is an introductory programming course. It forms the foundation for later CS courses, but it also satisfies Division V liberal studies credit. Upon successful completion of this course, a student should be able to do the following in the Java programming language:

- Solve programming problems through the use of conditionals, loops, and nested control structures
- Write an instantiable class from scratch
- Write code to call constructors and invoke methods on existing objects, including correct use of parameters and return values
- Demonstrate an understanding of commonly used operators (logical, arithmetic, and comparison)
- Demonstrate a basic working knowledge of arrays and their syntax

Evaluation of these learning outcomes will be done through written assessments (quizzes and/or exams).

Formal Communication Studies Requirement:

This course satisfies the Formal Communication Studies requirement. This course is designed to introduce students to the ways in which information and ideas are expressed using a communication system other than English. Such courses should foster the student's ability to conceptualize and communicate in an orderly, rational manner. Characteristics of a communication system include: 1) possession of a grammar; 2) operation from an established set of rules; 3) reasoning properties such as deduction, inference drawing and problem solving. This includes courses in languages and those in which the central focus of the course is on statistics, computers or formal logic.

Disability Services:

If you have a need for disability-related accommodations or services, please inform the Coordinator of Disability Services in the Dean of Students Office at 2001 C. B. Hedgecock Building (227-1700). Reasonable and effective accommodations and services will be provided to students if requests are made in a timely manner, with appropriate documentation, in accordance with federal, state, and University guidelines.