UNIVERSITY OF SASKATCHEWAN
Department of Computer Science
176 Thorvaldson Building
110 Science Place Saskatoon, SK S7N 5C9 Canada
Telephone: (306) 966-4886 Facsimile: (306) 966-4884

COURSE SYLLABUS
CMPT 140: INTRODUCTION TO CREATIVE COMPUTING

Catalogue Description:

Concepts in computing such as algorithms, problem solving, and programming are explored using interactive multimedia systems as the creative focus. Basic skills in problem solving, programming, design and interaction, event-based behaviour, and prototyping are developed.

Restrictions:

Recommended for students who do not have Computer Science 30. CMPT 140 can be taken for credit after the completion of CMPT 100, but CMPT 100 cannot be taken for credit after completion of CMPT 140. Students with credit for CMPT 105, CMPT 111, CMPT 113 or CMPT 116 cannot obtain credit for CMPT 140. CMPT 140 will count as a open elective for students in Interactive Systems Design, Computer Science, and Bioinformatics programs.

Prerequisite(s):	(none)
Class Time & Location:	Section 01 (Long): MWF 9:30-10:20, Arts 102
	Section 03 (Gutwin): TTh 13:00-14:20, Arts 143
Website:	moodle.cs.usask.ca

Instructor Information

Instructors:	Dr. Jeff Long	Dr. Carl Gutwin
Contact:	Email: jeff.long@usask.ca	gutwin@cs.usask.ca
	Office: Thorvaldson 178.7	Thorvaldson 377.2
	Office Phone: 966-4901	966-8646

Who is this course intended for?

This course is intended for students who are interested in Computer Science, but do not have CS30 (high school computer science). Students who already have CS30 or CMPT 105 are recommended to take CMPT 141. This course can be taken as a Science credit for all Arts & Science majors. If you are required to take CMPT 141 in your program, you can take 140 and 141 (in that order), complete your science requirements, and learn a highly valuable skill in the process!

Course Overview

We've designed this course to be a challenging and fun introduction to computing for students who have little or no background in computer science or programming (if you already have CS30 or CMPT 105, you should take CMPT 141). You'll start by learning to program using the Python language in Processing, a friendly, graphical, interactive environment. You'll be able to create programs that employ all the computational ideas in applications that emphasize graphics, animations, and interactive systems. As you progress through the course, you'll extend your understanding of computation using Python. You'll begin to see why it is used by scientists, scholars, software developers, and engineers. You'll be very well prepared to take CMPT 141, the next course, which also teaches computer science using Python.

Course Objectives

After completing this course, students will be able to:

- Design and implement simple Python programs from scratch.
- Test and debug simple Python programs.
- Employ conditionals and loops in simple Python programs.
- Employ variables, lists and dictionaries in simple Python programs.
- Define and call Python functions in Python programs.
- Trace through the execution of simple Python programs by hand.
- Implement simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max of a list.

Tutorial Schedule

The following tutorial sections are associated with this class

Section	Day	Time	Room
L01	M	12:30-13:50	Thorvaldson (Spinks) S320
L03	M	14:30-15:50	Thorvaldson (Spinks) S320
L05	T	10:00-11:20	Thorvaldson (Spinks) S320
L07	T	14:30-15:50	Thorvaldson (Spinks) S320
L09	W	10:30-11:50	Thorvaldson (Spinks) S311
L11	F	12:30-13:50	Thorvaldson (Spinks) S311

Student Evaluation

Grading Scheme

Lab Exercises (9 at 1% each)	9%
Assignments (9 at 4% each)	36%
Midterm Exam	15%
Final Exam	40%
Total	100%

Criteria that must be met to Pass

Students must write the final exam. A student who does not write the final exam will receive a grade of at most 49 in the course.

Attendance Expectation

- Attend every class, and participate actively. There is no penalty for missed lectures.
- Attend all laboratory (tutorial) sessions. These are opportunities to practice the course material with the guidance of a teaching assistant.
- Attend the midterm examination.

Midterm Exam

The midterm exam will be held outside of normal class times. This is because both class sections (01 and 03) will write the same midterm, at the same time. The midterm exam is scheduled for **Thursday, Oct. 27th from 5:00pm-6:30pm, in Arts 143**. Students should plan to make themselves available at that time. We will make accommodations for students who have a legitimate timing conflict.

Final Exam Scheduling:

The Registrar schedules all final examinations, including deferred and supplemental examinations. Students are advised not to make travel arrangements for the exam period until the official exam schedule has been posted.

Note: All students must be properly registered in order to attend lectures and receive credit for this course.

Textbook Information

Recommended Texts

- None

Topic Schedule

1. Algorithms and Computational Thinking
2. Abstraction and Encapsulation
3. Introduction to Processing/Pythong and Visual Output
4. Functions, part 1 (calling and defining)
5. Colour in Processing
6. Interaction and Events: The Interaction Cycle
7. Data, Expressions and Variables
8. Functions, part 2 (return values)
9. Libraries
10. Conditional Branching
11. Repetition
12. Nesting Constructs and Problem Solving
13. Lists
14. File I/O
15. Dictionaries
16. Advanced Problem Solving and Data Management

Policies

Recording of Lectures

Lecture videos will be available through a link on the course Moodle site.

Late Assignments

Late assignments will only be accepted under exceptional circumstances by the permission of the instructor.

Missed Assignments

Students are expected to attempt (and hopefully complete!) all assignments, and all laboratory exercises. It's better to submit partially completed assignments than to submit nothing at all. A missed assignment will receive a score of zero. If you miss an assignment for medical or compassionate reasons, contact your instructor as soon as possible.

Missed Examinations

1. Students who miss an exam should contact the instructor as soon as possible. If it is known in advance that an exam will be missed, the instructor should be contacted before the exam.

2. "A student who is absent from a final examination due to medical, compassionate, or other valid reasons, may apply to the College of Arts and Science Undergraduate Student's Office for a **deferred** exam. **Application must be made within three business days of the missed examination** and be accompanied by supporting documents." (http://artsandscience.usask.ca/students/help/success.php)

Incomplete Course Work and Final Grades

"*When a student has not completed the required course work, which includes any assignment or examination including the final examination, by the time of submission of the final grades, they may be granted an extension to permit completion of an assignment, or granted a deferred examination in the case of absence from a final examination.*

Extensions past the final examination date for the completion of assignments must be approved by the Department Head, or Dean in non-departmentalized Colleges, and may exceed thirty days only in unusual circumstances. The student must apply to the instructor for such an extension and furnish satisfactory reasons for the deficiency. Deferred final examinations are granted as per College policy.

In the interim, the instructor will submit a computed percentile grade for the class which factors in the incomplete coursework as a zero, along with a grade comment of INF (Incomplete Failure) if a failing grade.

In the case where the student has a passing percentile grade but the instructor has indicated in the course outline that failure to complete the required coursework will result in failure in the course, a final grade of 49% will be submitted along with a grade comment of INF (Incomplete Failure).

If an extension is granted and the required assignment is submitted within the allotted time, or if a deferred examination is granted and written in the case of absence from the final examination, the instructor will submit a revised assigned final percentage grade. The grade change will replace the previous grade and any grade comment of INF (Incomplete Failure) will be removed.

A student can pass a course on the basis of work completed in the course provided that any incomplete course work has not been deemed mandatory by the instructor in the course outline and/or by College regulations for achieving a passing grade." (http://policies.usask.ca/policies/academic-affairs/academic-courses.php)

For policies governing examinations and grading, students are referred to the Assessment of Students section of the University policy "Academic courses: class delivery, examinations, and assessment of student learning" (http://policies.usask.ca/policies/academic-affairs/academic-courses.php)

Academic Honesty

The University of Saskatchewan is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards regarding academic honesty and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Student Conduct & Appeals section of the University Secretary Website and avoid any behavior that could potentially result in suspicions of cheating, plagiarism, misrepresentation of facts and/or participation in an offence. Academic dishonesty is a serious offence and can result in suspension or expulsion from the University.

All students should read and be familiar with the Regulations on Academic Student Misconduct (http://www.usask.ca/secretariat/student-conduct-appeals/StudentAcademicMisconduct.pdf) as well as the Standard of Student Conduct in Non-Academic Matters and Procedures for Resolution of Complaints and Appeals (http://www.usask.ca/secretariat/student-conduct-appeals/StudentNon-AcademicMisconduct.pdf) Academic honesty is also defined and described in the Department of Computer Science statement on Academic Honesty (http://www.cs.usask.ca/students/academic-honesty/index.php).

For more information on what academic integrity means for students see the Student Conduct & Appeals section of the University Secretary Website at:
http://www.usask.ca/secretariat/student-conduct-appeals/forms/IntegrityDefined.pdf

Examinations with Disability Services for Students (DSS)

Students who have disabilities (learning, medical, physical, or mental health) are strongly encouraged to register with Disability Services for Students (DSS) if they have not already done so. Students who suspect they may have disabilities should contact DSS for advice and referrals. In order to access DSS programs and supports, students must follow DSS policy and procedures. For more information, check http://students.usask.ca/health/centres/disability-services-for-students.php, or contact DSS at 966-7273 or dss@usask.ca.

Students registered with DSS may request alternative arrangements for mid-term and final examinations. Students must arrange such accommodations through DSS by the stated deadlines. Instructors shall provide the examinations for students who are being accommodated by the deadlines established by DSS.

Student Supports

Student Learning Services (SLS) offers assistance to U of S undergrad and graduate students. For information on specific services, please see the SLS web site https://www.usask.ca/ulc/.

The Student and Enrolment Services Division (SESD) focuses on providing developmental and support services and programs to students and the university community. For more information, see the SESD web site http://www.usask.ca/sesd/.

Appendix: ACM-2013 Learning Outcomes

This course achieves the following learning outcomes (listed alphabetically) from the ACM-2013 Computer Science Curriculum Guidelines (https://www.acm.org/education/CS2013-final-report.pdf):

- Explain the characteristics and defining properties of algorithms and how they relate to machine processing. [Familiarity, C-P-1]
- Analyze simple problem statements to identify relevant information and select appropriate processing to solve the problem. [Assessment, C-P-2]
- Discuss the importance of algorithms in the problem-solving process. [Familiarity, SDF-AD-1]
- Create algorithms for solving simple problems. [Usage, SDF-AD-3]
- Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment, SDF-FPC-1]
- Identify and describe uses of primitive data types. [Familiarity, SDF-FPC-2]
- Write programs that use primitive data types. [Usage, SDF-FPC-3]
- Modify and expand short programs that use standard conditional and iterative control structures and functions. [Usage, SDF-FPC-4]
- Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing. [Usage, SDF-FPC-5]
- Choose appropriate conditional and iteration constructs for a given programming task. [Assessment, SDF-FPC-7]
- Write programs that use lists and maps. [Partial, SDF-FPC-7]
- Implement basic numerical algorithms. [Usage, AL-FDSA-1]