School of Applied Science and Management

CPSC128

Introduction to Object Oriented Programming I

Term: Fall

Number of Credits: 3

Course Outline

INSTRUCTOR: Kate Chatfield-Reed
E-MAIL: kchatfieldreed@yukonu.ca
OFFICE: A2806
PHONE: (867) 456-8563
TIME/DATES: M/W 5:30 pm – 7:00 pm (lectures) and M 4:00 pm – 5:30 pm (tutorial)
CLASSROOM: A2702
OFFICE HOURS: schedule by email

COURSE DESCRIPTION

The goal of CPSC 128 is to introduce the student to the basics of computer science and programming. To this end it covers: techniques, methods, and tools for systematic development and maintenance of software systems and documentation; basic algorithms and data structures; and fundamental concepts of object oriented programming. The bulk of the course is spent practicing program design in order to reinforce the fundamental concepts and constructs. Good programming practices are emphasized throughout, including: top-down design, modularization, code re-use, debugging techniques, and creating useful documentation. While the language of choice for this course is Python, the learning outcomes should be transferable to other programming languages.

COURSE REQUIREMENTS

Prerequisite(s): Math 11. While no previous programming experience is required, any such experience is helpful.

EQUIVALENCY OR TRANSFERABILITY

Receiving institutions determine course transferability. Find further information at:
https://www.yukonu.ca/admissions/transfer-credit

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

- produce an object-oriented (OO) analysis and design for a problem.
- apply the principles of class inheritance, composition, and association to construct hierarchies of new classes.
- use the components and constructs necessary to implement an OO program in efficient, reusable, extensible code.
- produce clearly written and well-documented code.

- evaluate programs through the careful application of appropriate testing techniques to assess their reliability and correctness
- document the analysis, design, implementation and testing of a program constructed using OO principles.

COURSE FORMAT

Weekly breakdown of instructional hours

This course will be delivered with the following breakdown per week: three hours of lecture and 1 hour of tutorial. Although it will vary from individual to individual, students should expect to spend 5 hours on course material outside of the classroom time (per week) on studying or completing assignments.

Delivery format

This course will be delivered in a face-to-face (in person) format. Students will be expected to access the YU online learning platform for additional material (Moodle).

EVALUATION

Assignments	40%
Quizzes	15%
Tutorials	15%
Final Exam	30%
Total	100%

The final grade of this course will be based on the following:

Assignments (40%): Individual coding assignments will be given that cover all the material in the course. Late assignments will be accepted up to one week past the deadline and subject to a 5% late penalty per day they are late. Students may in hand one late assignment without penalty (it still must be submitted within a week of the original deadline).

Quizzes (15%): Small weekly quizzes usually done on paper.

Tutorials: Weekly tutorials will reinforce concepts throughout the course with individual or group activities. The assessment will be varied and may include small coding submissions, written reflections, code annotations, or an oral explanation of your work.

Final Examination (30%): A comprehensive final examination will be given at the end of the course. Students will write the exam at the Whitehorse campus of Yukon University.

Code handed in for the assignments, quizzes, and final should always represent the students’ own work and should not be copied from other sources including other students (past or present), the internet, or an AI tool (such as ChatGPT).

COURSE WITHDRAWAL INFORMATION

The last date to withdraw without academic penalty is Oct. 31, 2025. Refer to the YukonU website for other important dates.

TEXTBOOKS & LEARNING MATERIALS

No textbook is required for this course. All the necessary content is provided either through the course website or online resources.

https://ttopper.github.io/CPSC128/

ACADEMIC INTEGRITY

Students are expected to contribute toward a positive and supportive environment and are required to conduct themselves in a responsible manner. Academic misconduct includes all forms of academic dishonesty such as cheating, plagiarism, fabrication, fraud, deceit, using the work of others without their permission, aiding other students in committing academic offences, misrepresenting academic assignments prepared by others as one's own, or any other forms of academic dishonesty including falsification of any information on any Yukon University document. The assignments cannot be solved using generative artificial intelligence programs (for example no ChatGPT).

Please refer to Academic Regulations & Procedures for further details about academic standing and student rights and responsibilities.

ACADEMIC ACCOMMODATION

Reasonable accommodations are available for students requiring an academic accommodation to fully participate in this class. These accommodations are available for students with a documented disability, chronic condition or any other grounds specified in section 8.0 of the Yukon University Academic Regulations (available on the Yukon University website). It is the student's responsibility to seek these accommodations by contacting the Learning Assistance Centre (LAC): LearningAssistanceCentre@yukonu.ca.

TOPIC OUTLINE

0. Course start-up.

Part I: Procedural programming

1. Introduction to computer science.

2. SIPO (sequence, input, processing, and output) programming.

3. Selection control structures.

4. Repetition control structures.

Part II: Object-based programming

5. Aggregate data types 1: Lists and strings.

6. Functions.

7. Text files.

8. Aggregate data types 2: Dictionaries.

Part III: Object-oriented programming

9. Object-oriented programming (OOP) 1: Encapsulation.

10. Object-oriented design (OOD).

11. Object-oriented programming (OOP) 2: Polymorphism and inheritance

12. Testing and documentation

Final exam – date set by administration and available on the website
https://www.yukonu.ca/programs/courses/cpsc-128