



Course Outline

Course Name: Programming Fundamentals (TECH 104)

Academic Period: 2021 - 2022

Faculty:

Faculty Availability:

Associate Dean:

Jonathan Kim

jonathan.kim@humber.ca

Schedule Type Code:

Land Acknowledgement

Humber College is located within the traditional and treaty lands of the Mississaugas of the Credit. Known as Adoobiigok [A-doe-bee-goke], the "Place of the Alders" in Michi Saagiig [Mi-Chee Saw-Geeg] language, the region is uniquely situated along Humber River Watershed, which historically provided an integral connection for Anishinaabe [Ah-nish-nah-bay], Haudenosaunee [Hoeden-no-shownee], and Wendat [Wine-Dot] peoples between the Ontario Lakeshore and the Lake Simcoe/Georgian Bay regions. Now home to people of numerous nations, Adoobiigok continues to provide a vital source of interconnection for all.

Equity, Diversity and Inclusion Statement

Humber College and the University of Guelph-Humber (Humber) are leaders in providing a learning, working and living environment that recognizes and values equity, diversity and inclusion in all its programs and services. Humber commits to reflect the diversity of the communities the College serves. Students, faculty, support and administrative staff feel a sense of belonging and have opportunities to be their authentic selves.

Faculty or Department	Faculty of Applied Sciences & Technology
Course Name:	Programming Fundamentals (TECH 104)
Pre-Requisites	none
Co-Requisites	none
Equates	none
Restrictions	Postsecondary; Faculty of Applied Sciences and Technology
Credit Value	3
Total Course Hours	56

Developed By:**Prepared By:****Approved by:**

Jonathan Kim



Humber Learning Outcomes (HLOs) in this course.

The HLOs are a cross-institutional learning outcomes strategy aimed at equipping Humber graduates with the employability skills, mindsets, and values they need to succeed in the future of work. To explore all the HLOs, please consult the [Humber Learning Outcomes framework](#).

Course Description

N/A

Course Rationale

Applications development using programming language(s) is now essential part of Electrical/Electronics/Computer/Telecommunications/Mechatronics and many other engineering and technology programs. The students learn the logic of programming using C as programming language tool, in this course. This fundamental of programming course leads to number of programming courses in the subsequent semesters. The students in Electrical Engineering Technology program learn the logic to program in this course and use this logic in subsequent semesters when they take the PLC courses. C/C++ are the languages of choice when programming at the hardware level or at the embedded system level.

Course Learning Method(s)

- Problem Based Learning (PBL)

Learning Outcomes

- Identify solutions of the problem based on analyzing a problem and using pseudo-code or graphical notation.
- Describe the difference between programming and scripting languages.
- Differentiate compiled programming from interpreted programming
- Analyze simple problems to identify the input data, processing requirements and the output data, based on specified data types and arithmetic operations.
- Use programming environment to implement structured code using available testing and debugging tools
- Use conditional and control structures to implement solutions to problems
- Use data storage in scalar variables and simple arrays
- Design simple user-defined functions for applications and development.
- Design structured programs using conditional statements, loops and functions
- Apply strings and simple string functions in practical applications
- Use standard features in C programming language to document program and code
- Demonstrate testing the implemented code for the acceptance to the requirements, while explaining the importance of program testing.

Assessment Weighting

Assessment	Weight
Test 3	25%
Hands-On Quizzes	15%

Assessment	Weight
Test 2	20%
Lab Assignments	25%
Test 1	15%
Total	100%

Modules of Study

Module	Course Learning Outcomes	Resources	Assessments
Introduction to Computer Architecture: integration of hardware, software and importance of operating systems	<ul style="list-style-type: none"> Identify solutions of the problem based on analyzing a problem and using pseudo-code or graphical notation. 	Class notes and Reference links	<ul style="list-style-type: none"> Test 1 Lab Assignments Hands-On Quizzes Test 3
Problem definition; Identification of logical solutions; Implementation of solutions	<ul style="list-style-type: none"> Identify solutions of the problem based on analyzing a problem and using pseudo-code or graphical notation. 	Class notes and Reference links	<ul style="list-style-type: none"> Test 1 Lab Assignments Hands-On Quizzes Test 3
Introduction to C programming language; Interpreted and compiled languages; Basic input and output functions	<ul style="list-style-type: none"> Describe the difference between programming and scripting languages. Differentiate compiled programming from interpreted programming 	Textbook (King): Chapter 2	<ul style="list-style-type: none"> Test 1 Lab Assignments Hands-On Quizzes Test 3
Formatted output; Different data types and storage space; Arithmetic and logical operators, real numbers, integers, and characters	<ul style="list-style-type: none"> Analyze simple problems to identify the input data, processing requirements and the output data, based on specified data types and arithmetic operations. Use programming environment to implement structured code using available testing and debugging tools 	Textbook (King): Chapter 3 and 4	<ul style="list-style-type: none"> Test 1 Lab Assignments Hands-On Quizzes Test 3

Module	Course Learning Outcomes	Resources	Assessments
Conditional Statements; Pseudo code for problem solution; Solution of problems using logical approach; Implementation of solution using C programming language	<ul style="list-style-type: none"> • Use programming environment to implement structured code using available testing and debugging tools • Use conditional and control structures to implement solutions to problems 	Textbook (King): Chapter 5	<ul style="list-style-type: none"> • Test 1 • Lab Assignments • Hands-On Quizzes • Test 3
Repetition statements; Pseudo code for solution of problems involving repetitive statements; Identifying logical solutions involving repetition	<ul style="list-style-type: none"> • Use conditional and control structures to implement solutions to problems 	Textbook (King): Chapter 6	<ul style="list-style-type: none"> • Test 1 • Lab Assignments • Hands-On Quizzes • Test 3
Arrays – one dimensional arrays; Arrays – initializing; Loops and arrays; Integer and real number processing using arrays; Reading and Writing character data in the arrays	<ul style="list-style-type: none"> • Use data storage in scalar variables and simple arrays 	Textbook (King): Chapter 8	<ul style="list-style-type: none"> • Lab Assignments • Test 2 • Hands-On Quizzes • Test 3
Functions; Definition and Calling functions; Design of structured programs; Local and Global function definitions and calls	<ul style="list-style-type: none"> • Design simple user-defined functions for applications and development. 	Textbook (King): Chapter 9	<ul style="list-style-type: none"> • Lab Assignments • Test 2 • Hands-On Quizzes • Test 3
Strings and simple string functions; Initialization of strings; Comparison of strings and character arrays; Simple string processing; String functions (e.g. strcmp, strlen, strcpy, strcat)	<ul style="list-style-type: none"> • Apply strings and simple string functions in practical applications 	Textbook (King): Chapter 13	<ul style="list-style-type: none"> • Lab Assignments • Test 2 • Hands-On Quizzes • Test 3

Module	Course Learning Outcomes	Resources	Assessments
Design of structured programs using conditional statements, loops and functions; Pseudo code solution for structured programs using conditional statements, loops and functions; Solution of problem using C programming language	<ul style="list-style-type: none"> • Use programming environment to implement structured code using available testing and debugging tools • Use conditional and control structures to implement solutions to problems • Design simple user-defined functions for applications and development. • Design structured programs using conditional statements, loops and functions • Use standard features in C programming language to document program and code • Demonstrate testing the implemented code for the acceptance to the requirements, while explaining the importance of program testing. 	Textbook (King): Chapters 5, 6, 8 and 9	<ul style="list-style-type: none"> • Lab Assignments • Test 2 • Hands-On Quizzes • Test 3

Required Resources

King, K. N. (2008) *C Programming, A Modern Approach*, second edition, W.W. Norton & Company, ISBN: 978-0-393-97950-3

Supplemental Resources

http://publications.gbdirect.co.uk/c_book/

Essential Skills

Section	Skills	Measurement	Details
Critical Thinking and Problem-Solving	<ul style="list-style-type: none"> • Analysing • Synthesizing • Evaluating • Decision-Making 	Teach and measure	<ul style="list-style-type: none"> • Skills will be taught throughout the semester with course material (lecture, hands-on labs, demonstration). • Summative and formative assessments (assignments, quizzes and tests).

Section	Skills	Measurement	Details
Personal Skills	<ul style="list-style-type: none"> Managing self Managing change and being flexible and adaptable Demonstrating personal responsibility 	Reinforce and measure	<ul style="list-style-type: none"> Skills will be reinforced throughout the semester with course material (lecture, hands-on labs, demonstration). Summative and formative assessments (assignments, quizzes and tests).
Communication	<ul style="list-style-type: none"> Reading Writing Visual Literacy 	Reinforce and measure	<ul style="list-style-type: none"> Lecture, labs. Summative and formative assessments

Prior Learning Assessment & Recognition (PLAR)

Prior Learning Assessment and Recognition (PLAR) is the formal evaluation and credit-granting process whereby candidates may obtain credits for prior learning. Prior learning includes the knowledge competencies and skills acquired, in both formal and informal ways, outside of post-secondary education. Candidates may have their knowledge, skills and competencies evaluated against the learning outcomes as defined in the course outline. Please review the [Assessment Methods Glossary](#) for more information on the Learning Portfolio assessment methods identified below.

The method(s) that are used to assess prior learning for this course may include:

- Challenge Exam (results recorded as a % grade and added to student's CGPA)
- Skills Test
- Interview

Please contact the Program Coordinator for more details.

Academic Regulations

It is the student's responsibility to be aware of the College Academic Regulations. The Academic Regulations apply to all applicants to Humber and all current students enrolled in any program or course offered by Humber, in any location. Information about academic appeals is found in the [Academic Regulations](#).

Anti-Discrimination Statement

At Humber College, all forms of discrimination and harassment are prohibited. Students and employees have the right to study, live and work in an environment that is free from discrimination and harassment. If you need assistance on concerns related to discrimination and harassment, please contact the [Centre for Human Rights, Equity and Inclusion](#) or the [Office of Student Conduct](#).

Accessible Learning Services

Humber strives to create a welcoming environment for all students where equity, diversity and inclusion are paramount. Accessible Learning Services facilitates equal access for students with disabilities by coordinating academic accommodations and services. Staff in Accessible Learning Services are available by appointment to assess specific needs, provide referrals and arrange appropriate accommodations. If you require academic accommodations, contact:

[Accessible Learning Services](#)

North Campus: (416) 675-6622 X5090

Lakeshore Campus: (416) 675-6622 X3331

Academic Integrity

Academic integrity is essentially honesty in all academic endeavors. Academic integrity requires that students avoid all forms of academic misconduct or dishonesty, including plagiarism, cheating on tests or exams or any misrepresentation of academic accomplishment.

Disclaimer

While every effort is made by the professor/faculty to cover all material listed in the outline, the order, content, and/or evaluation may change in the event of special circumstances (e.g. time constraints due to inclement weather, sickness, college closure, technology/equipment problems or changes, etc.). In any such case, students will be given appropriate notification in writing, with approval from the Dean (or designate) of the School.

Given the circumstances due to COVID-19, Humber reserves the right to alter the mode of delivery and examinations/assessments in this course.

Copyright

Copyright is the exclusive legal right given to a creator to reproduce, publish, sell or distribute his/her work. All members of the Humber community are required to comply with Canadian copyright law which governs the reproduction, use and distribution of copyrighted materials. This means that the copying, use and distribution of copyright-protected materials, regardless of format, is subject to certain limits and restrictions. For example, photocopying or scanning an entire textbook is not allowed, nor is distributing a scanned book.

See the [Humber Libraries website](#) for additional information regarding copyright and for details on allowable limits.

Humber College Institute of Technology and Advanced Learning • 2021/2022.