UNIVERSITY OF THE PHILIPPINES MINDANAO
College of Science and Mathematics
Department of Mathematics, Physics, and Computer Science

AMAT 152: Computer Programming I
1st Semester 2023-2024

COURSE GUIDE

COURSE DESCRIPTION

Elements of programming: design and analysis of algorithms and basic concepts in computer program development (control structures, arrays, structures, functions, text files, recursion, pointers, data structure).

COURSE LEARNING OUTCOMES

At the end of the course, students should be able to formulate solutions for a variety of problems through implementation of well-structured programs in the C/C++ Language. Specifically, the students should be able to:

1. Express formulated solutions as algorithms;
2. Demonstrate sound coding practices from their understanding of the basic concepts in program development such as declaration, input and printing, variables, operators, control statements, input, and output; and
3. Effectively implement their formulated solutions in the C/C++ language, making use of programming constructs including simple and structured data types, program control structures, arrays, loops, structures, pointers, functions, recursion, text files, and basic data structure.

COURSE OUTLINE

I. Elements of Programming
  A. Design of Algorithms
    a. Problem Solving with a Computer
    b. Introduction to Algorithm Development
    c. Representation of Algorithms
      i. Pseudocode
      ii. Flowchart

II. Implementation of Algorithms (Programming)
  A. Representation of Data
    a. Simple Data Types
    b. Structured Data Types
    c. Variable Declaration
    d. Characters and Strings Basics
  B. Statements, Expressions, and Operators
  C. Basic Program Control Structures
    a. Sequence
      i. Selection (if, else, else if)
    b. Repetition (for, while, do-while)
    c. Selection (switch)
    d. Loops
      i. Counter Controlled Loops
      ii. Event Controlled Loops
      iii. Infinite Loops
      iv. Continue and goto
  D. Arrays
    a. Single Dimension Arrays
    b. Multi Dimension Arrays

c. Initialization and Size of Arrays
d. Arrays as Pointers
e. Pointer Operations
f. More on char and strings

E. Structures
   a. Initialization using Structures
   b. Application of Structures
F. Functions and Structured Programming
   a. Function Declaration and Pass by Value
   b. Pass by Reference
   c. Recursion
G. File Manipulation

References:

- Roberts, Eric S. (1995). *The art and science of C: a library-based introduction to computer science*. Addison-Wesley
- Brookshear, J. Glenn. (1994). *Computer science: an overview*. Benjamin/Cummings Pub. Co.
- Deitel, P. and Deitel, H. (2015). *C: How to Program, 8th Edition*. Pearson

Related Resources:

- MIT Open Courseware: Lecture notes on C programming.
- Might be useful for first-time programmers: C programming tutorial
- Video: C Programming Crash Course

MODE OF DELIVERY

Learning Approach – LDM 3 (Classic Blended Learning)

This course will follow a blended-learning approach – consisting of face-to-face lecture and laboratory sessions and asynchronous activities. Course materials will be provided and can be accessed at any time in Canvas.

A study guide will be provided approximately five days before the topic will be discussed.

The lecture sessions of the week will involve a quiz/seatwork and discussion of the topic/s.

The laboratory session of the week will involve programming session and laboratory exercises.

For most of the time, students will learn the topics using the study guides and the learning resources, attend the sessions, perform in-class activities, practice their programming and problem-solving skills through laboratory exercises, and answer quizzes/exams. Students can access the materials through Canvas LMS.

Communication Plan

Communication will either be face-to-face or online.

Face-to-face communication will be either through face-to-face sessions or the faculty's consultation hours. For consultation, kindly request for appointment first to the faculty.

Meanwhile, online communication will be done asynchronously through Canvas or email.

MATERIALS NEEDED

The following are needed in this course:

- Desktop, laptop, tablet, or smartphone to access the learning materials and activities;
- UP Email account to register/log-in to Canvas, Zoom, and Google Classroom (for online mode of learning);
- Any PDF reader to view learning materials and activities;
- DevC++ v5.11, a C/C++ integrated development environment (IDE) for coding, debugging, and running your programs (laboratory exercises, long exams / machine problems). Instructions on how to install DevC++ are provided on the last page;
- MS Word or any equivalent word processor to type your laboratory exercises and long exams; and/or
- Image scanning app to scan your handwritten laboratory exercises and long exams.

STUDY SCHEDULE

[Note: Online Learning Resources (OLRs) links are embedded in the study guides]

Week	Session	Day	Topic/Activity	Activity Upload	Activity Due
1 (Aug 28 - Sept 1)	Lec Session 1.1	W			
	Lec Session 1.2	F	**Course Overview**	Study Guide 1 (Pseudocode and Flowchart)	
	Lab Session 1	W/F			
2 (Sept 4 - 8)	Lec Session 2.1 – 2.2	W	**Quiz/Discussion/Seatwork on** • Introduction to Algorithm Development • Algorithms • Pseudocode • Flowchart		Q
		F		Study Guide 2 (Variable Types and Constants)	SW
	Lab Session 2	W/F			
3 (Sept 11 - 15)	Lec Session 3.1 – 3.2	W	**Quiz/Discussion/Seatwork on** • Basic Computer Memory • Variables and Keywords in C/C++ • Numeric Variable Types • Variable Declarations • Constants • String Basics – Reading Strings from Keyboard		Q
		F		Study Guide 3 (Statements, Expressions, and Operators)	SW

	Lab Session 3	W/F	**Laboratory Session:** Pseudocode and Flowchart **Laboratory Exercise 1:** Pseudocode and Flowchart	LabExer1	LabExer1
4 (Sept 18 - 22)	Lec Session 4.1 – 4.2	W	**Quiz/Discussion/Seatwork on** • Statements and White Spaces • Null and Compound Statements • Expressions • Assignment Operators • Unary and Binary Mathematical • Basic Operators and Precedence • Relational and Logical Operators and • Precedence • Compound Assignment Operators • Conditional Operators and Precedence		Q
		F		Study Guide 4 and 5 (Program Control Statements)	SW
	Lab Session 4	W/F	**Laboratory Session:** The Basics **Laboratory Exercise 2:** Variable Types and Constants	LabExer2	LabExer2
5 (Sept 25 - 29)	Lec Session 5.1 – 5.2	W	**Quiz/Discussion/Seatwork on** • Program Control Statements • Sequential Control • Selection Control (single, two, three, multiple alternatives) • Dangling Else Statements • Repetition Control (for loops, do-while, while)		Q
		F		Study Guide 6 (Arrays)	SW
	Lab Session 5	W/F	**Laboratory Session:** Statements, Expressions, and Operators **Laboratory Exercise 3**: Statements, Expressions, and Operators	LabExer3	LabExer3
6 (Oct 2 - 6) - Online	Lec Session 6.1 – 6.2	W	**Quiz/Discussion/Seatwork on** • Naming and Declaring Arrays • Single and Multidimensional Arrays • Initialization of Arrays • Size of Arrays		Q
		F		Study Guide 7 (Pointers)	SW
	Lab Session 6	W/F	**Quiz/Discussion/Seatwork on** • More on Program Control Statements **Laboratory Session**: Program Control Statements **Laboratory Exercise 4**: Program Control Statements	LabExer4	LabExer4

7 (Oct 9 - 13)	Lec Session 7.1 – 7.2	W	**Long Exam 1**		
		F			
	Lab Session 7	W/F	**Laboratory Exam 1 (On-the-Spot Programming)**		
(Oct 16 – 21)			**Reading Break**		
8 (Oct 23-27)	Lec Session 8.1 – 8.2	W	**Quiz/Discussion/Seatwork on** • Arrays as Pointers • Pointer Manipulations		Q
		F		Study Guide 8 (Characters and Strings)	SW
	Lab Session 8	W/F	**Laboratory Session: Arrays and Pointers**		
9 (Oct 30-Nov 3)	Lec Session 9.1 – 9.2	W	*Holiday (All Saints' Day)*		
		F	**Quiz/Discussion/Seatwork on** • More on Characters and Strings • The char Data Type • ASCII Table • Manipulating strings • Characters and Strings	Study Guide 9 (Structured Programming and Functions)	SW
	Lab Session 9	W/F	**Laboratory Exercise 5**: Arrays	LabExer5	LabExer5
10 (Nov 6 - 10)	Lec Session 10.1 – 10.2	W	**Quiz/Discussion/Seatwork on** • Structured Programming and Functions		Q
		F	• Program flow with functions • Function Calls and Events • Writing a Function • Parameters and Arguments • Passing and Returning a Value • Recursion	Study Guide 10 (More on Functions)	SW
	Lab Session 10	W/F	**Laboratory Session**: Characters and Strings **Laboratory Exercise 6**: Characters and Strings	LabExer6	LabExer6
11 (Nov 13-17)	Lec Session 11.1 – 11.2	W	**Long Exam 2**		
		F			
	Lab Session 11	W/F	**Laboratory Exam 2 (On-the-Spot Programming)**		
12 (Nov 20-24) – Online Asynchronous	Lec Session 12.1 – 12.2	W	**Quiz/Discussion/Seatwork on** • Variable Scope • Variable Lifetime External Variables		Q
		F	• Local Variables • Static Variables	Study Guide 11	SW

			• Automatic Variables • Pass by Value • Pass by Reference • Passing Arrays to Functions	(Structures)	
	Lab Session 12	W/F	**Laboratory Session**: Structured Programming and Functions 1 **Programming Project Consultation**		
13 (Nov 27 - Dec 1) – Online Asynchronous	Lec Session 13.1 – 13.2	W	**Quiz/Discussion/Seatwork on** • Structures • Basic Structure Definition • Writing Structures and Structure Instances • Manipulating Structure Members • Complex Structures - Structures inside Structures		Q
		F		Study Guide (Files)	SW
	Lab Session 13	W/F	**Laboratory Session**: Structured Programming and Functions 2 **Programming Project Consultation** **Laboratory Exercise 7**: Structured Programming and Functions	LabExer7	LabExer7
14 (Dec 4-8)	Lec Session 14.1 – 14.2	W	**Quiz/Discussion/Seatwork on** • Introduction to Files • File Handling Functions • File Manipulation (Opening/Closing a File, Input/Output operations on files, Reading/Writing to files. • Differences in File Manipulation Modes		Q
		F			SW
	Lab Session 14	W/F	**Laboratory Session: Structures and Files** **Laboratory Exercise 8**: Structures and Files	LabExer8	LabExer8
15 (Dec 11-15)	Lec Session 15.1 – 15.2	W	**Long Exam 3**		
		F			
	Lab Session 15	W/F	**Laboratory Exam 3 (On-the-Spot Programming)**		
16 (Dec 19-20)	Lec Session 16.1	W	**Programming Project Consultation**		
Jan 3-10		**F2F**	**Programming Project Presentation**		

COURSE REQUIREMENTS

1. Lecture Component
    a. Quizzes/Seatwork – 10%
    b. Long Exams – 30%
2. Laboratory Component
    a. Laboratory Exercises – 20%
    b. Laboratory Exams – 25%
    c. Programming Project – 15%

Course Requirement 1a: Quizzes/Seatwork

A quiz is a short evaluation of what the students have learned conceptually while studying the course guide. This will be conducted before the discussion of the topic and will take 5-10 minutes long.

Seatwork will be done **during** the lecture session to further assess the problem-solving skills of the students after further discussion of the topic/s. These may be an individual or a group activity and may also involve group work presentation.

Course Requirement 1b: Long Exams

First part of the long examinations is concept-type, where students will be evaluated of their understanding on the theoretical concepts of programming. They will also consist of programming/flowchart questions to assess their algorithm-reading skills. There will be three long examinations in this course with their corresponding weights covering the following topics:

Long Exam 1 (15%)	Algorithm Design (Pseudocode and Flowcharts); on Basic Programming Elements (Variable declarations, Statement Declarations, Operators, Program Control Statements)
Long Exam 2 (15%)	Arrays, Characters, and Strings (single and multidimensional arrays, arrays as pointers, array and string manipulation); as well as past topics
Long Exam 3 (20%)	Structures, Functions, and File Handling (Declaration and initialization of structures, structure member manipulations. Complex structures, structured programming, implementing functions, function calls and events, parameters and arguments, recursion, file handling functions, and file manipulation); as well as past topics

Course Requirement 2a: Laboratory Exercises

Laboratory Exercises consist of programming problems aimed to apply the concepts learned during lecture and laboratory sessions into writing a program.

There will be 8 laboratory exercises in this course.

UNIVERSITY OF THE PHILIPPINES MINDANAO
College of Science and Mathematics
Department of Mathematics, Physics, and Computer Science

Course Requirement 2b: Laboratory Examinations

Laboratory Examinations are in a form of on-the-spot programming exams, where students are assessed on their overall programming skills in real-time by applying the theoretical and laboratory concepts and skills obtained throughout the course.

The following rubric will be used for evaluating outputs of the laboratory exercises and exams (Note: scores will be scaled, this is a sample of a distribution of points in a 100-point laboratory program output):

CRITERIA	RATINGS			POINTS
Program Specification (Will be credited if and only if the program runs and the criterion can be checked)	**40 pts** **(100%)** All program requirements are satisfied.	**24 pts** **(60%)** One program requirement has not been satisfied.	**8 pts** **(20%)** At least two program requirements have not been satisfied.	**40 points**
Program Execution (Will be credited if and only if the program runs and the criterion can be checked)	**20 pts** **(100%)** Program runs correctly without any error.	**12 pts** **(60%)** Program runs with one or two errors found.	**4 pts** **(20%)** Program runs with more than two errors found.	**20 points**
Comments/ Documentation	**20 pts** **(100%)** Comments are well-written and clearly explains section of the code is accomplishing and how.	**12 pts** **(60%)** Comments are terse but still give a brief overview of what the whole code accomplishing and how.	**4 pts** **(20%)** Vague/inappropriate to few comments	**20 points**
Coding Style	**10 pts** **(100%)** Code is well-written, with consistent style, and easy to follow.	**6 pts** **(60%)** Coding style is inconsistent.	**2 pts** **(20%)** Code is either hard to follow or incomprehensible.	**10 points**
Program Additional	**10 pts** **(100%)** Program scores perfect in the other requirements and it exceeds the specifications.	**6 pts** **(60%)** Program is only according to the specifications but still scored perfect in the other requirements.	**2 pts** **(20%)** Program is only according to the specifications.	**10 points**
				Total Points: 100

UNIVERSITY OF THE PHILIPPINES MINDANAO
College of Science and Mathematics
Department of Mathematics, Physics, and Computer Science

Course Requirement 2c: Programming Project

This is a culmination of the students' learning in this course. Students will be formed into groups of 5 and will develop a program that solves a real-world problem of interest.

Groupings will be generated as earliest as possible. Consultations will open during laboratory sessions to make sure that the groups are making progress.

The expected outputs are the following:

- Source file (.c)
- Report paper (.pdf) presenting the problem and the solution that was made in a form of pseudocode/flowchart with algorithm summary, describing how the program runs (in detail). The paper should include the following:
    - Title Page
    - The Problem – background, objectives, significance
    - The Program - description/summary of the whole algorithm, user input/output, program pseudocode/s or flowchart/s, others
- A group programming project presentation and demonstration during final exam schedule

The same rubric above will be used to score the C program.

Project Criteria:

- Program – 80%
- Report Paper – 10%
- Program Presentation and Demonstration – 10%

C Program Requirements

The program must contain the following:

- at least one program control structure (selection, repetition)
- at least one array, string, and pointer
- at least one user-defined function (aside from the main() function)
- at least one structure
- a file input/output

More information will be provided regarding the project when necessary.

GRADING SCALE

[98, 100)	1.00
[94, 98)	1.25
[90, 94)	1.50
[85, 90)	1.75
[80, 85)	2.00
[75, 80)	2.25
[70, 75)	2.50
[65, 70)	2.75
[50, 65)	3.00
[0, 50)	5.00
	DRP

*Accurate to 4 decimal places. Grade of 4.0 and INC will **not** be given.

CLASS ACADEMIC POLICIES

Academic Dishonesty. All forms of academic dishonesty will not be tolerated. Any student caught committing any form of academic dishonesty will undergo investigation. If found guilty of cheating, s/he/they will receive a final grade of 5.0 for the course. Refer to Section 2 and Section 25 of the UP Diliman Faculty Manual.

Exams and Quizzes. No make-up quizzes/seatwork will be given whether excused or unexcused. For missed exams, at most one make-up exam will be given for students whose absences are excused as certified by the OCS. Missing more than one exam would incur the student a score of zero (0) for the next missed exam/s. **Complaints regarding the checking of exams/quizzes will only be entertained within one week of its release.** Dates of examination stated above are subject to changes.

Deadlines. Deadlines are strictly followed. A grace period will be provided for seatwork or laboratory exercises **when necessary**. For laboratory exercises, the grace period will be until 11:59 pm on that same day the exercise was given. For seatwork, the grace period will be before the start of the laboratory session. Non submission of the week's seatwork/laboratory exercises beyond the grace period will result in a score of zero (0) for that missed submission. Complaints regarding the checking of the problem sets/seatwork will be entertained only up to the next class meeting after its release.

Attendance. As per university rule, a student must not absent him/herself more than 20% of the total number of classes for the term. The maximum number of allowable excused/unexcused absences is six (6) lecture sessions and three (3) laboratory sessions. A student who exceeded these allowable absences will be FORCED to drop from the course and will merit a final grade of DRP if and only if majority of the absences are excused with proper document proof duly approved by the Office of the College Secretary; else the student will get a grade of 5.0.

It is important that students submit their excuse slip/letter (if applicable) as soon as possible after being absent. Failure to do this is a sign of irresponsibility.

Dropping. A student is officially dropped from the course if he/she/they successfully completes the dropping process. A student who stops attending classes and/or taking exams without officially dropping will be given a grade of 5.0. Note that the teacher has the right not to sign the dropping slip.

ABOUT THE INSTRUCTOR

In this semester, the faculty-in-charge for AMAT 152 is Ms. El Veena Grace A Rosero.

Ms. El Veena Grace Rosero (Ma'am Grace) is an Instructor from the Department of Mathematics, Physics, and Computer Science. Her research interests include nature-inspired optimization and mathematical modelling. She uses the following computer programming languages in solving mathematical and simulation problems:

- C
- MATLAB
- Python
- R

Students can contact her through Canvas or the following:

- Email - earosero@up.edu.ph (for admin-related purposes only)
- Cellular number - +639095636453 (TNT) or +639271630200 (Globe)

CLASS/HOUSE RULES

1. Be mindful of time. As much as possible,
   - Learn the course topics according to the given schedule;
   - Keep updated with the announcements/reminders regularly;
   - Submit outputs on time; and
   - Turn up to class on time.
2. Fulfill your role as a learner (and a teacher to your peers):
   - Read the study guides; read/watch the learning resources; practice solving problems provided to you;
   - Participate in any activities and discussions as much as possible;
   - Comply the course requirements with the best of your ability. Avoid submitting sloppy work.
3. Please communicate only on the channels and venues specified by your professors.
4. When communicating with your professors in any form of media or setting, please be courteous and forward your queries with clarity. To learn more about netiquette, kindly read this blog:
5. Please follow email etiquette when communicating with professors via email. Kindly read this blog to know more about email etiquette.
6. UPhold Honor and Excellence. Always remember that UP values
   - the rightful use of learning resources provided in this course, which is included in the intellectual property rights. Students are not allowed to distribute any copyrighted educational-use-only learning materials outside this class.
   - academic honesty and integrity. Any form of academic dishonesty (like cheating and plagiarism) by the student will be sanctioned with disciplinary action in addition to an automatic grade of 5.00.
7. Never forget to read and follow instructions. Failure to do so will result in point deductions.
8. You are encouraged to seek help from other peers in other courses, as well as on other professors teaching AMAT 152. If you encounter any difficulties in the conduct of the course, we will try to help as much as we can. Make use of the faculty's consultation hours! Our call is for everyone to finish this course with honor, integrity, and excellence.
9. If you have a medical or any other condition that could get in the way of your optimum performance in class, please inform the instructor right away so s/he can address the matter in a timely and appropriate manner and comply the necessary documents for your absence to be excused.

Instructions on installing DevC++ plus additional resources

- C/C++ Tutorial:
  https://www.w3schools.com/cpp/
- Dev-C++ v5.11 Download:
  http://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/Dev-Cpp%205.11%20TDM-GCC%204.9.2%20Setup.exe/download
- Dev-C++ Resources
  - http://ocw.uc3m.es/ingenieria-informatica/programming-in-c-language-2013/IntroductiontoDevCIDE.pdf
  - https://www.bloodshed.net/dev/doc/index.html
  - http://www.cplusplus.com/doc/tutorial/introduction/devcpp/

  These are especially useful for those who have little or no programming experience.