CS103
Computer Programming 2

Lecture 0: Course Orientation

Bicol University College of Science
CSIT Department
2nd Semester, 2023-2024

Course Description

This course is a continuation of CS102-Computer Programming 1. This course allows the student to apply advanced techniques in procedural programming to solve computational problems adhering to the standards and guidelines of documentation. It covers the creation, manipulation and application of user-defined data structures (arrays, matrices, records, and linked-lists), recursion, and file-handling techniques.

Course Learning Outcomes

- Design, implement, test, and debug programs to solve a set of computing problems, based on given specifications, by applying their knowledge of:
  - fundamentals of programming,
  - recursions,
  - intermediate data structures (arrays, matrices, records, and linked-lists),
  - files.
- Apply an appropriate coding standard with proper documentation in order to come up with a readable, understandable and maintainable program code.
- Select and apply appropriate debugging and testing techniques to ensure the correctness of a program and compliance to given specifications, to solve a computing problem.

Course Content

A. Recursion

- *Format of recursive Functions*
- *Simulation of Recursive Functions*
- *Tracing using Recursive Tree*

B. Arrays

- *Defining Arrays*
- *Array Examples*
- *Passing Arrays to Functions*
- *Sorting Arrays*
- *Searching Arrays*
- ***Multiple-Subscripted Arrays*** *(Matrices)*

C. Strings

- *Fundamentals of Strings & Characters*
- *Character Arrays*
- *String Manipulations*

MIDTERM EXAM

D. Pointers

- *Pointer Variable Definition and Initialization*
- *Pointer Operators*
- *Pointer Arguments to Function by Reference*
- *Const Values and Parameters*
- *sizeof Operator*
- *Comparing Pointers*
- Arrays of Pointers

E. Structures

- *Uses of Structures*
- *Declaring and Defining Structures*
- *Declaring and Defining Nested Structures*
- *Passing Structures to a Function*
- *Self-Referential Structures*
- *Dynamic Memory Allocation*
- ***Linked Lists***

FINAL EXAM

Course Requirements

- Major Exams
- Quizzes
- Programming Exercises
- Practical Examination
- Programming Project
- Class Participation/Recitation

Grading System

Your final mark is first calculated as a percentage according to the following weights:

• Major Exams(Midterm, Final)	35%
• Quizzes	25%
• Practical Examination	15%
• Programming Project	15%
• Class Participation/Programming Exercises/ Recitation	10%

The percentage grades are transmuted and then translated into numerical grades between 1.0 – 5.0 according to the standard translation table used by the university.

Grading System

Outstanding: 1.0-99-100%; 1.1-98%; 1.2-97%; 1.3-96%; 1.4-95%

Superior: 1.5-94%; 1.6-93%; 1.7-92%

Very Satisfactory: 1.8-91%; 1.9-90%; 2.0-89%; 2.1-88%; 2.2-87%; 2.3-86%; 2.4-85%

Satisfactorily: 2.5-84%; 2.6-82-83%; 2.7-80-81%

Fair: 2.8-78-79%; 2.9-76-77%; 3.0-75%

Conditional Failure – (Lowest possible mid-term rating): 4.0
Failure: 5.0
Incomplete: INC
Dropped: DRP

Academic Honesty

- Every student is expected to uphold the principles of academic and intellectual honesty.
- Plagiarism and other forms of cheating are considered to be serious academic offenses.
- If a student is caught cheating on an exam, he or she will be given zero mark for the exam. If a student is caught cheating twice, the student will be referred to the Office of Student Affairs and Services(OSAS) and be given a failing grade in accordance with the guidelines as stated in the BU Students Handbook.

Policies on Absences and Tardiness

- Attendance is not part of your grade but you are still expected to attend and participate in the classes.

- According to CHED policy, total number of absences by the students should not be more than 20% of the total number of meetings. Students incurring more than 20% of the total number of meetings of unexcused absences automatically gets a failing grade regardless of class standing.

Use of the FB GC and BULMS

- Facebook messenger group chat will be used for instant communication and announcements.
  - Observe the given guidelines on how to behave in the GC
- BU Learning Management System will be utilized for posting and/or submission of learning resources and activities such as assignments(programming exercises, project), quizzes and exams.

Policies on Quizzes, Exams

- No make-up quizzes
- Special/make-up exam will be available for students who missed the midterm/final exam for medical reasons and can document this with a doctor's note or for extraordinary circumstances beyond the control of the student.
- Request for special/make-up exam must be made prior or immediately after the conduct of the exam. No make-up exam will be given once the result is returned already.

Programming Exercises , Practical Exam

- Programming exercises are given to assess if the students were able to apply the concepts discussed in the lecture.

- Practical examination will be given during the midterm exam period

Programming Project

- It will be a collaboration of 5 members
- Start formulating ideas for your problem/topic. The problem/topic must be approved first before you can start working on it.
- It should have at the least, an application of file handling, arrays and structures.
- Source code, documentation, presentation/defense

Caveat

- There maybe adjustments or changes that will be made throughout the semester.
- Students are responsible to learn about these changes if they miss any class time.
- Issues and matters not stated in this orientation that are important for this course should be consulted with the course Instructor.

Using the Lecture Presentations

Not everything on the slides will be read aloud by the speaker/teacher

Not everything the speaker says will be on the slides

You need both! And then some...

Beyond the Lectures

You won’t learn programming by just listening to the lectures.

What you need in addition is:

- The ability to ask questions and get them answered
- A textbook/reference you can use for details and examples not in the lectures
- And most important: hands-on practice

Resources

REFERENCES:

- **Paul Deitel, and Harvey Deitel, “C How to Program with an introduction to C++” Eight Edition**
- **Kimberly Nelson King, “C Programming: A Modern Approach” Second Edition**
- **Brian W. Kernighan and Dennis M. Ritchie, “C Programming Language” 2nd Edition**
- **Jeri R. Hanly and Elliot B. Koffman, “Problem Solving and Program Design in C”, 8th Edition**

Metzler, N. (2019). *C for Beginners: An Introduction to Learn C Programming with Tutorials and Hands-On Examples*. Lightbulb Publishing

Prinz, P. & Crawford, T. (2016). *C in a Nutshell: The Definitive Reference, 2nd Edition*. O’Reilly Media, Inc., CA, U.S.A.

- Farell, J. (2015). *Programming Logic and Design, Comprehensive, 8th Edition*. Cengage Learning Asia Pte. Ltd., Singapore
- Klemens, B. (2015). *21st Century C: C Tips from the New School, 2nd Edition*. **O’Reilly Media, Inc., CA, U.S.A.**

Kochan, S.G. (2015). *Programming in C, 4th Edition (Developer's Library)*. Pearson Education, Inc., New York, N.Y.

Perry, G. & Miller, D. (2014). *C Programming Absolute Beginner’s Guide, 3rd Edition*. Pearson Education, Inc., New York, N.Y.

Hanly, J.R. & Koffman, E.B. (2012). *Problem Solving and Program Design in C*, 7th Edition”. Pearson Education, Inc., New York, N.Y.

Forouzan, B.A. & Gilberg, R.F. (2011). *Foundations of C Programming*. Cengage Learning Asia Pte. Ltd., Singapore

SUGGESTED READINGS

Frey, C.B. (2019). *The Technology Trap: Capital, Labor, and Power in the Age of Automation*. Princeton University Press, U.S.A.

Erwig, M. (2017). *Once Upon an Algorithm: How Stories Explain Computing*, The MIT Press.

Spinellis, D. (2016). *Effective Debugging: 66 Specific Ways to Debug Software and Systems (Effective Software Development Series) 1st Edition*, Addison-Wesley Professional.

Advising and Support

- Email: aasatuito@bicol-u.edu.ph
- Consultation schedule is by appointment