



# **ICS-103**

## **Computer Programming in C**

### **Lecture 1**

### **Course Introduction**

**Dr. Tarek Ahmed Helmy El-Basuny**

# Outline of Today's Class

## □ We will introduce the course and agree on some coordination issues.

- ⇒ Contact data and setting the office hours,
- ⇒ Course Information,
- ⇒ References of the Course,
- ⇒ Notes for Class Attendance,
- ⇒ Course Outline: Topics to be covered during this course,
- ⇒ Course objective and learning outcomes,
- ⇒ Grading Policies,
- ⇒ Advices for achieving high grade in the course,
- ⇒ Quick Review of:
  - What is a computer program?
  - Imperative programming
  - Declarative programming
  - Similarity between a text book and a computer program
  - Computer's HW and SW

## Contact Information

- My name is: Tarek Ahmed Helmy El-Basuny
- My email address is: [helmy@kfupm.edu.sa](mailto:helmy@kfupm.edu.sa)
- My office location: Building # 22, Room #: 137-3,
- My office Tel # is: 1967
- Office hours will be on:
  - Sunday, Tuesday From 9:00 to 10:00 AM
  - Monday and Wednesday from 2:00 to 3:30 PM
- If you have any commitments during the above days and time then you may arrange for a meeting by phone or email.

## Course Information

- ❑ **Course Name:** Computer Programming in C.
- ❑ **Course Code:** ICS 103
- ❑ Sections 9, 10, & 11 offering times are: MW, from 8: to 8:50 am, then 9:00 to 9:50 am and 10:00 to 10:50 am, all in Building 24-120.
- ❑ **Co-requisites:** MATH 101 or MATH 132.
- ❑ **Note: ICS 103 cannot be taken by ICS/SWE students**
- ❑ It is 3 credit hours course, where:
  - 2 lectures of 50 minutes each per week will be offered.
  - 1 lab of 3 hours per week will be conducted.
  - FYI, there will be no lab in the first week.
- ❑ It is a tightly coordinated course. The material, Programming Assignments, Midterm and Final Exams are common for all sections.
- ❑ Only quizzes will be different per section.

## References of the Course

- **Main textbook is:** Problem Solving and Program Design in C, 8<sup>th</sup> Ed., by Jeri R. Hanly & Elliot B. Koffman. Prentice Hall, 2016.
- **Topics presented in the classes should be given high priority.**
- **Lab documents.**
- **It is recommended to read other C programming-related Websites and books to improve your programming skills.**
- **Other Recommended References:**
  - C How to Program, 7/e, By Deitel & Deitel, 2010.
  - C for Engineers and Scientists, by Harry H. Cheng. McGraw Hill, 2009.
  - C Programming: A Modern Approach, 2/e, by K. N. King, 2008.
  - Engineering Problem Solving with C, 3/e, by Delores M. Etter, 2005.

## Important Notes for Attendance

- Regular attendance is a University requirement.
- Attendance will be taken at the **beginning/end** of the classes **by circulating the attendance sheets**. I will allow you to attend in any of my sections if you can't attend in your section's time for any reason.
- **Late arrivals** disrupts the class session. Hence, two late attendances (**more than 10 minutes**) will be considered as one absence.
- Missing more than 6 unexcused lectures or 3 unexcused labs will result in a DN grade without prior warning.
- **Unexcused absence** will become an **excused absence ONLY** by an official letter from the **Dean of student's affair office**.
- Assignments must be submitted through the blackboard before the due date.
- **No** late or email submission will be accepted.

# Topics Outline

- ❑ Week # 1, Overview of Computers and Programming,
- ❑ Week # 2, Overview of C language (i.e. Variable Declarations, Data Types, General Form of a C program, etc.)
- ❑ Week # 3, Arithmetic Expressions: Formatting Numbers in the Program Output, Common Programming Errors,
- ❑ Week # 4, Selection Structures (i.e. if-then, if-then-else, switch-case),
- ❑ Weeks # 5, Repetition in the program (i.e. for loop, while loop, do-while loop),
- ❑ Week # 6, Building Programs from Existing Library Functions, Top-Down Design and Structure of Programs,
- ❑ Week # 7: Functions with and without Arguments,
- ❑ Week # 8, Pointers and Modular Programming,
- ❑ Week # 9, Data Files: **Text files** (organizes data as consecutive characters, i.e. text files) & **Binary files** (organizes data into blocks containing contiguous bytes of information, i.e. arrays, etc.), **File Operations** (i.e. Creating a new file, Opening an existing file, Closing a file, Reading from and writing information to a file),
- ❑ Week # 10, Arrays, Array Arguments, Using Array Elements as Function Arguments,
- ❑ Week # 11, Searching and Sorting an Array, Parallel Arrays and Enumerated Types,
- ❑ Week # 12, Multidimensional Arrays, and Array Processing,
- ❑ Weeks # 13 & 14, Strings,
- ❑ Week # 15, Review and Problem Solving.

## Course Objective

- ❑ Provide engineering students with basic knowledge of problem solving and programming in C language.
- ❑ Improve students programming and problem-solving skills through projects carried out in the lab and HWs.



## Course Learning Outcome

- ❑ After completing this course, you should be able to:
  - Understand and find the output of simple C programs that incorporate different types of variables, expressions (**arithmetic and logical**), selection, and iteration.
  - Understand and find the output of more complex C programs containing arrays and invoking/**calling** functions having input and output arguments using pointers.
  - Design and implement simple programs using basic syntax of C language **such as assignment, expressions, selection, and iterations.**
  - Practice modular programming by developing more complex C programs made **of functions passing data between them using arrays, input, and output arguments.**

## Grading/**Assessment** Policy

<b>Class Assessment:</b> to be handled by the Class Instructor	<b>Weight</b>
HW Programming Assignments (4 HWs, 1.25 % each)	5 %
Quizzes, (5 quizzes 4% each) ( <b>during the classes and to TBA</b> )	20 %
<b>Midterm Exam 1:</b> Wed. Oct. 31 <sup>st</sup> (7:00 – 9:00 PM), <b>Location, TBA.</b> <b>Material:</b> from the first set of lecture slides up to (and including) all of repetition: Chapters 1, 2, 3, 4, and 5.	25%
<b>Final Exam: 20% Comprehensive,</b> December 17, 2018 at 7:00 PM, <b>Location (TBA).</b>	30%
<b>TOTAL</b>	<b>80 %</b>

<b>Lab Assessment:</b> to be handled by the Lab Instructor	<b>Weight</b>
Attendance + Lab work	2.5 %
Lab Tests	14.5 %
Final Test	3 %
<b>TOTAL</b>	<b>20 %</b>

## Advices for better Performance in the Course

- ❑ We would like the course to run smoothly and to be enjoyable.
  - Help each other to understand the fundamentals of C Programming.
  - Be supportive and collaborative with me.
  - Feel free to ask about any ambiguous/unclear concepts during the classes.  
Do not wait to the next class.
- ❑ Feedback, by email or in person, always welcome.
  - Feel free to let us know what you find just good or bad about the course.
- ❑ I am really in your side and doing my best to help you.
  - Do not misuse the democratic way of interaction.
  - I can handle criticism or difference of opinions.
- ❑ Let us apply active learning strategies.

## Who wants to pass this course with A+?

- ❑ Of course you all seek that and this means you have to :
  - ➡ Attend the lectures and the labs
  - ➡ Concentrate and give attention to everything explained in the classes and labs.
  - ➡ Do the homework's and assignments by yourself.
  - ➡ Try to practice coding and debugging the programs by yourself, **do not depend on others.**
  - ➡ Repeat the lectures examples yourself.
  - ➡ and most importantly:

**Read with understanding the slides and the Text Book**



# What is a Computer Program?

- ❑ A computer program is a sequence of instructions written to perform a specified task with a computer.
- ❑ A program/**source code** is written in a programming language that usually follows one of two main paradigms: **imperative** or **declarative** programming.
  - **Imperative programming (i.e. C or Java)** is a paradigm in which the programmer **defines sequences of commands for the computer to perform**.
  - **Declarative programming (i.e. Prolog)** is a paradigm in which the programmer **expresses** what the program should do **without** prescribing how to do it in terms of sequences of actions to be taken.
- ❑ **Source code** file will be converted into an **object/machine code** by the **compiler** and later converted into an **executable** file (**after being linked with other library functions**) to be executed by a CPU.

## Similarity between a Program and a Book

- ❑ The structure of a **computer program** is similar to the structure of a **Book**.
- ❑ Do you think so?
- ❑ How is that?

## Similarity between a Program and a Book

□ Quick review of what a **Book** looks like:

- A book objective is **to help the reader learn something**.
- A book consists of a number of **chapters**.
- Each **chapter** consists of a number of **paragraphs**.
- Each **paragraph** consists of a number of **sentences**.
- Each **sentence** must obey the **syntax rules** in the **writing language**.
  - An **ordinary sentence** follows the syntax: **subject verb ...**
  - A **question (sentence)** follows syntax: **verb subject ...**

## Similarity between a Program and a Book

□ Quick review of what a C program looks like:

- A C Program has an objective, to solve a certain problem.
- A C Program consists of a number of Functions and Variables.
- Each Function consists of a number of statements.
- Each statement must obey the syntax rules in the C language.
- One statement can only perform a simple operation.
- One statement is insufficient to solve a complex problem.
- A complex task can be performed by many statements.





**The End!!**

**Thank you**

**Any Questions?**