



# Course Specification

— (Bachelor)

**Course Title:** Problem Solving in Computing

**Course Code:** CSC 1102

**Program:** Bachelor in Computer Science

**Department:** Department of Computer Science

**College:** Computers and Information Technology

**Institution:** University of Tabuk

**Version:** 1.0

**Last Revision Date:** 27 July 2022



## Table of Contents

<b>A. General information about the course:</b> .....	3
<b>B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods</b> .....	4
<b>C. Course Content</b> .....	6
<b>D. Students Assessment Activities</b> .....	7
<b>E. Learning Resources and Facilities</b> .....	7
<b>F. Assessment of Course Quality</b> .....	8
<b>G. Specification Approval</b> .....	9



## A. General information about the course:

### 1. Course Identification

1. Credit hours: ( 3 )

#### 2. Course type

A.  University  College  Department  Track  Others  
B.  Required  Elective

3. Level/year at which this course is offered: (First Semester / First Year)

#### 4. Course general Description:

This course will present basic concepts of problem solving and how problems are analyzed and solved on computers. The course describes logical concepts required for ordering and creating solutions that can be effectively implemented as algorithms running on computers and designing appropriate solutions.

#### 5. Pre-requirements for this course (if any):

NA

#### 6. Co-requisites for this course (if any):

NA

#### 7. Course Main Objective(s):

After completing this course, the students will be able to:

- 1) Understand the problem-solving concepts in computing.
- 2) Describe the problem-solving steps to solve general computational problems.
- 3) Set up and evaluate expressions and equations using variables, constants, operators, and the hierarchy of operations.
- 4) Use abstraction, decomposition, sorting and searching techniques for analyzing and solving computational problems.
- 5) Use algorithms, flowcharts, and pseudocode to develop the instructions for modularized computational solution.
- 6) Explain and apply the logic and control structures (sequential, decision, and loops).

### 2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom		



No	Mode of Instruction	Contact Hours	Percentage
2	E-learning	60	100%
3	Hybrid <ul style="list-style-type: none"> <li>Traditional classroom</li> <li>E-learning</li> </ul>		
4	Distance learning		

### 3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	30
2.	Laboratory/Studio	30
3.	Field	
4.	Tutorial	
5.	Others (specify)	
<b>Total</b>		<b>60</b>

## B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
<b>1.0</b>	<b>Knowledge and understanding</b>			
1.1	Identify problem solving and computational thinking as a concept of computing	ILO1, ILO 2	Lectures Class Discussions	Exams Assignments
1.2	Describe different control structures used in solving problems related to computer programs	ILO1, ILO 2	Lectures Class Discussions	Exams Assignments
1.3	Discuss problem in terms of a planned solution	ILO1, ILO 2	Lectures Class Discussions	Exams Assignments



Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.4	Clarify notions and definitions in program modularization	ILO1, ILO 2	Lectures Class Discussions	Exams Assignments
<b>2.0</b>	<b>Skills</b>			
2.1	Implement analytical solutions to solve problems	ILO 2	Lectures Class Discussions Case Study	Exams Assignments Projects
2.2	Appraise the importance of logic, decisions, and loops in solving problems	ILO 2	Lectures Class Discussions Case Study	Exams Assignments Projects
2.3	Acquire the basic programming skills to implement solutions using suitable data types and constructs	ILO 2	Lectures Class Discussions	Exams Assignments
2.4	Develop thought skills such as sorting, searching and pattern recognition for solving computer problems	ILO 2	Lectures Class Discussions Case Study	Exams Assignments Projects
2.5	Use abstraction, decomposition and modularization as a computational thinking practice	ILO 2	Lectures Class Discussions	Exams Assignments Projects
<b>3.0</b>	<b>Values, autonomy, and responsibility</b>			
3.1	Demonstrate self-learning and continuing professional development	ILO9	Class Discussions Case Study	Class Participation Presentations
3.2	Use communication and teamwork skills	ILO9, ILO10	Class Discussions Case Study	Class Participation Presentations
3.3	Practice computer ethics effectively and professionally	ILO2	Class Discussions Case Study	Class Participation Presentations



## C. Course Content

No	List of Topics	Contact Hours
1	<p><b>General Problem-Solving Concepts: Problem Solving in Everyday Life, Types of Problems, Problem Solving in Computing, Challenges in Problem Solving</b></p> <p><b>Practical: practice suggesting problems, defining their types and classifying them; and use the six problem-solving steps to solve a simple computing problems</b></p>	4
2	<p><b>Abstraction and Decomposition: expressing ideas in specific contexts while suppressing irrelevant details</b></p> <p><b>Practical: apply abstraction to different problems</b></p>	4
3	<p><b>Decomposition: Breaking a complex problem into smaller and more manageable parts/steps</b></p> <p><b>Practical: apply decomposition to different problems</b></p>	4
4	<p><b>Solution Planning: Communicating with Computer, Organizing Solution, Testing Solution, Coding Solution and Algorithms, Software Development Cycle</b></p> <p><b>Practical: practice the use of: problem analysis chart, Interactivity Chart, IPO (input-processing-output) chart and algorithms</b></p>	4
5	<p><b>Solution Planning: Flowchart Symbols, Pseudocode, UML Overview</b></p> <p><b>Practical: Use algorithms, flowchart and pseudocode to solve simple problems and convert pseudocode into an algorithm</b></p>	4
6	<p><b>Introduction to Programming Languages: Constants and Variables, Data Types, Operators, Mathematical Functions , Expressions and Data Structure, Pointers for Structuring Solution, Modules and Methods, Local and Global Variables, Parameters and Return Values</b></p> <p><b>Practical: Practice writing a pseudocode code for implementing constants, variables, operators, assignment statements, mathematical functions, local variables, global variables and calling methods</b></p>	4
7	<p><b>Problem Solving with Decisions and Loops: Decision Logic Structure, Multiple If/Then/Else Instructions, Using Straight-Through Logic, Using Positive and Negative Logic, Logic Conversion</b></p> <p><b>Practical: writing pseudocode and algorithms; and drawing flowcharts for implementing decision logic with if/then, if/then/else and nested ifs using different types of logic</b></p>	4
8	<p><b>Problem Solving with Decisions and Loops: Loop Logic Structure, While/While End and Repeat/Until loops</b></p> <p><b>Practical: writing pseudocode and algorithms; and drawing flowcharts for implementing loop logic structure using while/while end and repeat/until loops</b></p>	4
9	<p><b>Problem Solving with Decisions and Loops: Automatic-Counter Loop</b></p>	4



	<b>Nested Loops, Indicators, Recursion</b> <b>Practical: writing pseudocode and algorithms; and drawing flowcharts for implementing automatic-counter loop and nested loops</b>	
10	<b>Arrays: One-Dimensional Arrays, Two-Dimensional Arrays, Multidimensional Arrays</b> <b>Practical: writing pseudocode and drawing flowcharts for using arrays to solve related problems</b>	4
11	<b>Types of Searching: Linear and Brute Force Search, Binary Search</b> <b>Practical: apply linear search, brute force search, and binary search using case studies</b>	4
12	<b>Types of Sorting: General and Internal Sorting Algorithms</b> <b>Practical: apply sorting algorithms using case studies</b>	4
13	<b>Stacks and Queues: Overview of Stack and Queue data structures</b> <b>Practical: use queue and stack in solving problems</b>	4
14	<b>Pattern recognition: Looking for similarities among and within problems, which also enable reuse knowledge of previous similar problems</b> <b>Practical: apply pattern recognition to different problems</b>	4
15	<b>Application of computational thinking techniques (Applications in different domains, for example, financial data computing, puzzle solving, development of games, and scientific computation)</b> <b>Practical: apply a case study</b>	4
<b>Total</b>		<b>60</b>

#### D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	In-Class Discussion and Participation	2-16	10%
2.	Assignments	2-16	20%
3.	Project and presentations	6 or 7	20%
4.	Mid-Exam	6 or 7	20%
5.	Final Exam	16 or 17	30%

\*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

#### E. Learning Resources and Facilities

##### 1. References and Learning Resources





<b>Essential References</b>	Maureen Sprankle, Jim Hubbard, 2011. Problem Solving and Programming Concepts 9th Edition. ISBN-10 : 0273752219, ISBN-13 : 978-0273752219
<b>Supportive References</b>	Karl Beecher, 2017, Computational Thinking - A beginner's guide to problem-solving and programming, 1st Edition, BCS Learning & Development Limited, ISBN: 9781780173641 Walter Savitch, 2017. Java An Introduction to Problem Solving & Programming, 8th Edition. ISBN-10 : 0134462033, ISBN-13 : 978-0134462035 Walter Savitch & Kenrick Mock, 2017 "Problem Solving with C++", Pearson; 10th edition, ISBN-10 : 0134448286, ISBN-13 : 978-0134448282
<b>Electronic Materials</b>	
<b>Other Learning Materials</b>	

## 2. Required Facilities and equipment

Items	Resources
<b>facilities</b> (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Fully equipped computer Lab
<b>Technology equipment</b> (projector, smart board, software)	- Projector with HDMI port - Operating System: Windows - Software: Java, C++, Python, C#
<b>Other equipment</b> (depending on the nature of the specialty)	

## F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of Teaching	Faculty, Program Leaders, and Advisory Board	Both Direct and Indirect
	Students	Indirect
Effectiveness of Students Assessment	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Both Direct and Indirect
Quality of Learning Resources	Faculty, Students, and Advisory Board	Indirect
The Extent to which CLOs have been Achieved	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Direct (as in section B) and Indirect/Surveys
	Students	Indirect



Assessment Areas/Issues	Assessor	Assessment Methods
Other	-	-

**Assessors** (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify))

**Assessment Methods** (Direct, Indirect)

### G. Specification Approval

<b>COUNCIL /COMMITTEE</b>	
<b>REFERENCE NO.</b>	
<b>DATE</b>	

