# Fundamentals of Programming

## Module Information

| | | | |
|---|---|---|---|
| **Module Title** | **Fundamentals of Programming** | **Module Delivery** | |
| **Module Type** | **Core** | ☒ **Theory**<br>☒ **Lecture**<br>☒ **Lab**<br>☐ **Tutorial**<br>☐ **Practical**<br>☒ **Seminar** | |
| **Module Code** | CYBT103 | | |
| **ECTS Credits** | 5 | | |
| **SWL (hr/sem)** | 125 | | |
| **Module Level** | 1 | **Semester of Delivery** | 1 |
| **Administering Department** | Cyber Security Techniques Engineering | **College** | Technical Engineering College for computer and AI / kirkuk |
| **Module Leader** | | **e-mail** | E-mail |
| **Module Leader's Acad. Title** | | **Module Leader's Qualification** | |
| **Module Tutor** | | **e-mail** | |
| **Peer Reviewer Name** | | **e-mail** | |
| **Scientific Committee Approval Date** | 31/12/2025 | **Version Number** | 1.0 |

## Relation with other Modules

| | | | |
|---|---|---|---|
| **Prerequisite module** | | **Semester** | 2 |
| **Co-requisites module** | | **Semester** | |

## Module Aims, Learning Outcomes and Indicative Contents

| | |
|---|---|
| **Module Objectives** | 1. Introducing the fundamentals and principles of programming in C++ language.<br>2. Teaching the concept of Procedure Oriented Programming.<br>3. Starts from scratch to advance programing by improving the skills of the students through several program implementation and code writing.<br>4. The students should be able to define programming purposes & the required code lines to perform the needed works.<br>5. as well as qualifying him to use the different kinds of programming style and program functions in building & executing the projects of cyber security engineering. |
| **Module Learning Outcomes** | 1. Understanding the fundamentals of programming in C++ language.<br>2. Mastering C++ programming tools and techniques, including common integrated development environment (IDE) such as visual studio.<br>3. Becoming familiar with the C++ concepts such as Variables, assignments, Simple input, Main program and functions.<br>4. Being competent in common If-statement,Loops, Boolean Expressions & Logical operators.<br>5. Being able to perform Function call, Parameters, return values.<br>6. Being able to write C++ codes, Basics of program design & Programming style. |

| **Learning and Teaching Strategies** | |
|---|---|
| **Strategies** | The main strategy that will be adopted in delivering this module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. This will be achieved through classes, interactive tutorials and by considering types of simple experiments involving some sampling activities that are interesting to the students. |

| **Student Workload (SWL)** | | | |
|---|---|---|---|
| **Structured SWL (h/sem)** | 63 | **Structured SWL (h/w)** | 4 |
| **Unstructured SWL (h/sem)** | 62 | **Unstructured SWL (h/w)** | 4.1 |
| **Total SWL (h/sem)** | **125** | | |

## Module Evaluation

| | | Time/Number | Weight (Marks) | Week Due |
|---|---|---|---|---|
| Formative assessment | Quizzes | 4 | 10% (10) | 3,6,9,12 |
| | Assignments | 8 | 5% (5) | Every other week |
| | Projects / Lab. | 14 | 10% (10) | Continuous |
| | Report | 2 | 5% (5) | 7, 14 |
| | Seminar | 1 | 10% (10) | 15 |
| Summative assessment | Midterm Exam | 2hr | 10% (10) | 7 |
| | Final Exam | 3hr | 50% (50) | 16 |
| Total assessment | | | 100% (100 Marks) | |

## FUNDAMENTALS OF PROGRAMMING - PROGRAMME COURSE DESCRIPTION

| Code CYBT 103 | Name of the Course Unit | Semester | In-Class Hours (T+P) | Credit | ECTS Credit |
|---|---|---|---|---|---|
| | Fundamentals of Programming | 1 | 2+3 | | 5 |

| GENERAL INFORMATION | |
|---|---|
| Language of Instruction : | English |
| Level of the Course Unit : | BACHELOR'S DEGREE |
| Type of the Course : | Compulsory |
| Mode of Delivery of the Course Unit | Face to Face |
| Coordinator of the Course Unit | |
| Instructor(s) of the Course Unit | |

| OBJECTIVES AND CONTENTS | |
|---|---|
| Objectives of the Course Unit: | Introducing the fundamentals and principles of programming in C++ language. Starts from scratch to advance programing by improving the skills of the students through several program implementation and code writing. The students should be able to define programming purposes & the required code lines to perform the needed works as well as qualifying him to use the different kinds of programming style and program functions in building & executing the projects of cyber security engineering. |
| Contents of the Course Unit: | 1. Procedure Oriented Programming<br>2. Structure of a program<br>3. Flow Chart.<br>4. Variables, assignments, Simple input, Main program If-statement If-else, Boolean Expressions & Logical operators |

| OBJECTIVES AND CONTENTS | |
|---|---|
| | 5. Output formatting.<br>6. Pointers.<br>7. Dynamic Memory. |

| | Delivery Plan (Weekly Syllabus) |
|---|---|
| **WEEK** | **KEY LEARNING OUTCOMES OF THE COURSE UNIT (On successful completion of this course unit, students/learners will or will be able to)** |
| 1 | Introduction to C++ (Structure of a program), Flow Chart. |
| 2 | Variables, assignments, Simple input, Main program. |
| 3 | If-statement If-else, Boolean Expressions & Logical operators. |
| 4 | Loops Nested Loops, and program Design. |
| 5 | Output formatting. |
| 6 | Functions, Parameters, return values. |
| 7 | Debugger. |
| 8 | Lists Methods, Nesting, Slicing, and Comprehension. |
| 9 | Strings and String Formatting. |
| 10 | Dictionary and Handle Exceptions. |
| 11 | Values and references. |
| 12 | Basics of program design & Programming style. |
| 13 | Pointers (Reference operator, dereference operator, Declaring variables of pointer types, Pointers and arrays, Pointers to pointers, void pointers, and Pointers to functions). |
| 14 | Dynamic Memory (Operators new, check if the allocation memory is successful and Operators delete). |
| 15 | review/seminar |
| 16 | Final Exam. |

| Delivery Plan (Weekly Lab. Syllabus) | |
|---|---|
| | **Material Covered** |
| **Week 1** | Lab 1: Getting started - Structure of a program- Flow Chart. |
| **Week 2** | Lab 2: Variables, assignments, Simple input, Main program - Examples and Problems |
| **Week 3** | Lab 3: If-statement If-else, Boolean Expressions & Logical operators - Examples and Problems. |
| **Week 4** | Lab 4: Loops Nested Loops, and program Design - Examples and Problems. |
| **Week 5** | Lab 5: Loops Nested Loops, and program Design - Examples and Problems. |
| **Week 6** | Lab 6: Functions, Parameters, return values - Examples and Problems. |
| **Week 7** | Lab 7: Debugger - Examples and Problems. |
| **Week 8** | Lab 8: Lists Methods, Nesting, Slicing, and Comprehension - Examples and Problems. |
| **Week 9** | Lab 9: Dictionary and Handle Exceptions - Examples and Problems. |
| **Week 10** | Lab 10: Pointers - Examples and Problems. |
| **Week 11** | Lab 11: Strings and String Formatting - Examples and Problems. |
| **Week 12** | Lab 12: Programming style. |
| **Week 13** | Lab 13: Values and references. |
| **Week 14** | Lab 14: Dynamic Memory - Examples and Problems. |
| **Week15** | Lab 15: Review |
| **Week 16** | **Final Exam** |

| WORKLOAD & ECTS CREDITS OF THE COURSE UNIT<br>Fundamentals of Programming | | | |
|---|---|---|---|
| **Workload for Learning & Teaching Activities** | | | |
| **Type of the Learning Activates** | **Learning Activities (# of week)** | **Duration (hours, h)** | **Workload (h)** |
| Lecture & In-Class Activities | 15 | 2 | 30 |
| Preliminary & Further Study | NA | NA | NA |
| Land Surveying | NA | NA | NA |
| Group Work | NA | NA | NA |
| Laboratory | 15 | 3 | 45 |
| Reading | 6 | 1 | 6 |
| Assignment (Homework) | 8 | 2 | 16 |
| Project Work | NA | NA | NA |
| Seminar | 1 | 1 | 1 |
| Internship | NA | NA | NA |
| Technical Visit | NA | NA | NA |
| Web Based Learning | 6 | 2 | 12 |
| Implementation/Application/Practice | NA | NA | NA |
| Practice at a workplace | NA | NA | NA |
| Occupational Activity | NA | NA | NA |
| Social Activity | NA | NA | NA |
| Thesis Work | NA | NA | NA |
| Field Study | NA | NA | NA |
| Report Writing | 2 | 2 | 4 |
| Final Exam -Theory | 1 | 3 | 3 |
| Final Exam - Practical | 1 | 1 | 1 |
| Preparation for the Final Exam- Theory | 1 | 20 | 20 |
| Preparation for the Final Exam -Practical | 1 | 15 | 15 |
| Mid-Term Exam - Theory | 1 | 2 | 2 |
| Mid-Term Exam - Practical | 1 | 1 | 1 |
| Preparation for the Mid-Term Exam | 1 | 15 | 15 |
| Short Exam (Quizzes) | 4 | 0.5 | 2 |
| Preparation for the Short Exam (Quizzes) | 4 | 2 | 8 |
| **Total Workload of the Course Unit** | | | 175 |

## Learning and Teaching Resources

| | **Text** | **Available in the Library?** |
|---|---|---|

| | | |
|---|---|---|
| **Required Texts** | 1. **Choudhary, H. (2013). C++ Programming-Final Golden Edition. Beginners To Experts Approach Guide-With Easy Learning & Problem Analysis to Program Design & Development.** | no |
| **Recommended Texts** | 2. **Farrell, J. (2008). Object-oriented programming using C++. Cengage Learning.**<br>3. **Object-Oriented Programming in C++, Fourth Edition.** | no |
| **Websites** | 4. **https://www.geeksforgeeks.org/cpp**<br>5. https://www.w3schools.com/cpp | |

## Grading Scheme

| Group | Grade | Marks % | Definition |
|---|---|---|---|
| **Success Group (50 - 100)** | **A -** Excellent | 90 - 100 | Outstanding Performance |
| | **B -** Very Good | 80 - 89 | Above average with some errors |
| | **C -** Good | 70 - 79 | Sound work with notable errors |
| | **D -** Satisfactory | 60 - 69 | Fair but with major shortcomings |
| | **E -** Sufficient | 50 - 59 | Work meets minimum criteria |
| **Fail Group (0 – 49)** | **FX –** Fail | (45-49) | More work required but credit awarded |
| | **F –** Fail | (0-44) | Considerable amount of work required |
| | | | |

**Note:** Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.