**MINISTRY OF EDUCATION**

**OF AZERBAIJAN REPUBLIC**

**Azerbaijan State Oil and Industry University**

**SYLLABUS**

**Approved:** Doctor of Science in Mathematics, Prof. A.R. Aliyev

(head of department)

**Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date:** February 12, 2021

**Department**: General and Applied Mathematics

**Faculty**: Information Technologies and Management

**I. Information on the Subject**

**Subject**: **Fundamentals of programming - II**

**Academic load (total hours): 60**

lectures 30

seminars 30

**Academic year:** 2020 - 2021

**Semester : 2**

**Specialty 050509 – Computer Science**

**Number of credits: 6**

**II. Information on the Instructor**

**Sevinj R. Karimova , Senior Lecturer**

(surname, name, patronymic, academic title, academic degree)

**Consultation days and hours:** every week on Tuesday at 14.00

**E-mail:** kerimovasevinc66@gmail.com

**Work phone:** +99412 4986592 (urban)

**III. Required Textbooks and Manuals**

**Basic**:

1. Brian W. Kernigan , Dennis M. Ritchie , The C programming language. Second edition.

2. Stephen G. Kochan, Programming in C. A complete introduction to the C

programming language. Third Edition.

3. Kenneth C. Louden and Kenneth A. Lambert “Programming Languages:

Principles and Practice, Third Edition”

4. Herbert Schildt. ANSI C - Made Easy. Publisher: Osborne McGraw-Hill.

ISBN: 0-07-881500-2

5. Bjarne Stroustrup ,Programming. Principles and Practice Using C++, Addison –

Wesley, 2016.

6. Rod Stephens (2013) Essential Algorithms: A Practical approach to Computer

Algorithms, John Wiley & Sons, Inc.

**Supplementary**:

1. C Tutorial. www.tutorialspoint.com/cprogramming/

2. Learn C Language. <http://www.learncpp.com/>

3. C++ Tutorial. <http://www.tutorialspoint.com/cplusplus/>

4. https://www.studytonight.com/c/

*Additional materials, articles and book chapters will be provided by instructor.*

**IV. Objective and Description of the Subject**

**Brief description of the course**:

The academic aim of the course is to learn students C/C++ language and how to code effectively to solve problems from the real life.

C is a general-purpose, procedural, imperative computer programming language, developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language, which is also equally popular and most widely used among modern software programmers.

The students get a basic familiarity with programming using C programming language. They will learn basic data types, input and output, conditional and loop statements, how to work with linear and two dimensional arrays, numeric and character arrays, pointers and references. Students will be equipped with the knowledge and skills how to code basic algorithms using C language.

**Objective of the course**:

1) to develop computational thinking

2) to develop an understanding of the main principles of solving

problems using computers

Upon successful completion of this course, students will be able to:

* Understand the basic terminology used in computer programming
* Write, compile and debug programs in C language.
* Use different data types in a computer program.
* Design programs involving decision structures, loops and functions.
* Use different data structures to process data.

The syllabus aims to encourage the development of computational thinking, that is thinking about what can be computed and how by the use abstraction and decomposition. Upon successful completion of this course, the student will be able to understand concept in the design of programming languages and implement programs in different types of languages.

**V. List of Lecture Topics**

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic and Brief Description | Lecture | Seminars |
| 1 | **Topic № 1.**  Basic definitions of C/C++ programming language.  **Contents:**  Input and output.  Primary(Built-in) Data Types. Data types and type modifiers.  Bitwise Operators.  Operators in C/C++ programming. | 2 | 2 |
| 2 | **Topic №2.**  **Pointers.**  Addresses, & operator, ***%p*** format. Introducton to pointers. Pointers and arrays. Pointer arithmetic.  C/C++ **memory management.**  **Contents:**  Dynamic memory allocation with ***new*** and ***delete***. Applications in array processing. References.  Allocating Memory Dynamically.  malloc() function, calloc() function, etc. | 2 | 2 |
| 3 | **Topic №3.**  **Functions,** function **declaration.**  **Scope rules. local** variables. **global** variables.  **formal**  parameters. Ways to pass arguments in functions: pass by value and pass by reference.  The **call by value** method of passing arguments to a function .  **Contents:**  Function definition. Formal and actual parameters.  Function Prototypes. Argument Promotion Rules. | 2 | 2 |
| 4 | **Topic №4.** Arrays. Declaring Arrays. Accessing Array Elements. Multidimensional Array. Two-dimensional array. Relationship Between Pointers and Arrays. **Contents:**  Examples Using Arrays.  Initializing an Array in a Declaration with an Initializer List.  Specifying an Array’s Size with a Constant Variable and Setting. | 2 | 2 |
| 5 | **Topic №5.**  **Passing Arrays to Functions**  **Contents:**  Way 1:Formal parameters as a pointer.  Way 2: Formal parameters as a sized array.  Way 3: Formal parameters as an unsized array. | 2 | 2 |
| 6 | **Topic № 6.**  **Recursion. Examples of recursion.**  **Contents:**  Example Using Recursion: **Fibonacci** Series  **Factorial** function. Greatest common divisor.  **Towers** of **Hanoi**.  Recursion vs. Iteration | 2 | 2 |
| 7 | **Topic № 7.**  **Char arrays. Two dimentional arrays.**  **Contents:**  Char arrays: how to read, write and process. Two and multi dimentional arrays.  How to read, process and print two dimentional arrays. | 2 | 2 |
| 8 | **Topic № 8.** Searching Arrays. **Contents:**  Binary Search Algorithm.  Linear Search Algorithm.  Search, insert and delete in an unsorted array | 2 | 2 |
| 9 | **Topic № 9.** Sorting Arrays. **Contents:** Selection Sort, Insertion Sort, Bubble sort,Merge Sort , Quick Sort, Cocktail ( shaker ) Sort. | 2 | 2 |
| 10 | Topic № 10.  Structures. Arrays of structures.  **Contents:**  Accessing Structure Members. Structure as a function argument. | 2 | 2 |
| 11 | Topic № 11.  Input and output in C++.  **Contents:**  File system. Binary and text files.  Contents:  Binary files. Text files. | 2 | 2 |
| 12 | **Topic № 12.**  **Functions for working with files.**  **Contents:**  File Input /Output in C.  fopen(), fclose(), fscanf(), fprintf(), fseek() and so on. |  |  |
| 13 | **Topic № 13.**  Introduction to Object Technology.  Object-oriented programming (OOP)  Basic Object Technology Concepts.  **Contents:**  Encapsulation. Encapsulation asone of the **fundamental**  concepts in object-oriented programming(OOP). Inheritance. Type of Inheritance. Data Abstraction  Polymorphism  Inheritance |  |  |
| 14 | **Topic № 14.**  **Classes. Classes and Objects.**  Class Scope and Accessing Class Members.  **Contents:**  Object. Class . Methods .  Instance Variables .  Member Functions and Classes.Instantiation.  Classes. Properties and methods.  Access Specifiers.  Defining a Member Function. |  |  |
| 15 | **Topic № 15.**  **Constructors and Destructors.**  **Contents:** Parameterized Constructor.The Class Destructor. C++ Copy Constructor  Passing Arguments to Constructors.  **Overloading Constructors.** |  |  |

**Slides Presentation Themes**

1. Introduction to C++ Standard Library. Class Template vector.

## 2. Fundamentals of Operator Overloading.

3. Abstract Classes and Pure Virtual Functions (C++) .

4. Different Types of Inheritance (C++).

5. Friend Functions and Friend Classes (C++).

6. Object-Oriented Programming: Polymorphism.

7. Overview of the **Elm** programming language.

8. Overview of the **Visual Basic** programming language.

9. Overview of the **CoffeeScript** programming language.

10. Overview of the **PHP** programming language.

11. Overview of the **Lua** programming language.

### 12. Overview of the Ruby programming language.

13. Review of the programming language **Julia**.

14. An overview of the **Clojure** programming language.

15. Overview of the **Go** programming language.

16. Overview of the **Erlang** programming language.

**VI**. **Form of the Exam – in writing, orally, in the form of a dialogue or text**

The examination will be conducted in writing+**orally.**

**VII. Evaluation during the semester and points layout Maximum number of points**: 100 points

**A) Maximum number of points earned during the semester**: 50points

A midterm exam and a final exam account for a significant portion of the grade.

**B) According to the results of the semester exam:** maximum 50 points

Each examination card has 5 questions; for each question a maximum of 10 points are given

**Remark**: The number of points earned by the student at the examination shall not be less than 17.

**C) Evaluation on the results of the semester (based on the points earned at and before the examination):**

|  |  |  |
| --- | --- | --- |
| 91 – 100 points | Excellent | A |
| 81 – 90 points | Very good | B |
| 71 – 80 points | Good | C |
| 61 – 70 points | Satisfactorily | D |
| 51 – 60 points | Passed | E |
| less than 51 points | unsatisfactorily | F |

**Instructor**: **Signature**: \_\_ **Karimova Sevinj Rafig** \_\_\_\_\_\_\_\_

(surname, name, patronymic) **Date**: February 12, 2021