SYLLABUS

Discipline: Programming Principles I
Number of credits: 4
Term: Fall 20__
Instructors full name: Askar Akshabayev, Beisenbek Baisakov, Bobur Mukhsimbayev

Personal Information about the Instructor	Time and place of classes		Contact information
	Classes	**Office Hours**	**e-mail**
Askar Akshabayev	According to the schedule	Room 262, will be appointed	askar.akshabayev@gmail.com
Beisenbek Baisakov	According to the schedule	Room 272, will be appointed	beysenbek@gmail.com
Bobur Mukhsimbayev	According to the schedule	Room 262, will be appointed	bobur.muhsimbaev@gmail.com

COURSE DURATION: 3 credits, 15 weeks, 60 class hours

COURSE DESCRIPTION
This course is designed to introduce students to Procedure Oriented Programming concepts on the assumption that they are not familiar with programming. Its main aim is to teach principle of programming using C++ rather than attempting to give complete exposition of all the features of C++.

COURSE OBJECTIVES
The objective of this course is to provide the student with the fundamental knowledge and skills to become a proficient C++ programmer.

COURSE OUTCOMES
Students will be exposed to basic hardware and software concepts and familiar with issues related to software design. They will master using key structured programming constructs: declarations, sequence, selection, repetition, evaluating expressions, be familiar with using C++ functions and the concepts related to good modular design. They will learn working with one-dimensional, two-dimensional arrays, C++ structures, pointers and reference parameters. Also they will be familiar with using text file input/output.

COURSE POST REQUISITES
Knowledge and skills obtained during study of course Programming Languages are used in following courses: Programming Technologies, Object-Oriented Programming, Algorithms and Data Structure.

LITERATURE
1. C++ How to Program, Fifth Edition, H. M. Deitel, P. J. Deitel - Deitel & Associates, Inc., Prentice

Hall.
2. C++ for Dummies 5th Edition, Stephen Randy Davis, Wiley Publishing, Inc.
3. Practical C++ Programming, Steve Oualline, O'Reilly & Associates, Inc.
4. C++: The Complete Reference third edition, Herbert Schildt, McGraw-Hill

Week	Class work				Laboratory works
	Topic	Lecture	Practice	Chapters for reading	
1	**Introduction to C++:** ● Introduction to code structure ● Compiling and executing program ● Introduction to data types ● Representing Numbers: int, double, float ● Comments ● Introduction to git, piazza	2	2	**[1]** 1.1, 1.2, 1.3, **[2]** 1.1, **[3]** 1.1, 1.2, 1.4 **[4]** 1.3, 1.5	*Laboratory work #1*
2	**Variable and Data Types:** ● Introduction to numeric systems ● freopen ● Math library functions ● Introduction to Char, String ● Operators and Operands, value casting ● Unary Operators ● Binary operators ● Bit Manipulations	2	2	**[1]** 2.1, 2.2, 2.3, **[2]** 1.3, 1.5, **[4]** 2.1, 2.2, 3.4 **[1]**2.6, **[2]** 1.12, **[3]** **[4]** 3.1-3.3	*Laboratory work #2*
3	**Conditional Statements:** ● Logical Operators (and, or, xor, not) ● Logical Comparisons ● If else statement (nested if else statements)	2	2	**[1]** 4.1, **[2]** 1.8, 1.9, **[3]** **[4]** 3.6, 3.7 **[1]** 4.2, **[2]** 2.8, **[4]** 2.4	*Laboratory work #3*
4	**Loop operators:** ● Loop code structure ● For, while, do while statements ● Dead loop, continue, break	2	2	**[1]**4.2, **[2]**2.1, 2.4, **[3]** **[1]** 4.2, **[2]** 2.2, **[4]** 2.8, 3.9	*Laboratory work #4.*

5	**Nested Loop statements:** ● Types of nested loops ● Nested while loop ● Nested for loop ● Syntax of Nested loops	2	2	**[1]** 2.1, **[2]** 3.1-3.4 **[4]** 5.1 **[1]** 2.2, **[2]** 3.5, **[4]** 6.1, 6.2	*Laboratory work #5*
6	**Arrays:** ● What is Arrays ● Types of Arrays ● Array declaration ● Accessing element of array ● Searching In Array ● Bubble Sort ● Arrays as parameters to function	2	2	**[1]**2.2,2.3, **[2]** 3.7,3.9, 3.12, **[4]** 5.4, 6.3-6.5 **[1]** 3.1, **[2]** 4.8, **[4]** 7.1, 7-7.3	*Laboratory work #6*
7	**Two-Dimensional arrays:** ● Initializing Two-Dimensional Arrays ● Accessing Two-Dimensional Array Elements ● Multidimensional arrays	2	2	**[1]** 2.4, 3.1, 3.2, **[2]** 3.8, 3.11, 5.2-5.6, **[4]** 5.5, 5.6, 7.5, 7.6 **[1]** 2.5, **[2]** 3.14- 3.16, **[4]** 5.3, 6.9	*Laboratory work #7*
8	*Midterm*	2	2	**[1]** 2.7, **[2]** 6.1-6.5, 7.1, 7.2,	
9	**String functions:** ● library string ● arrays of characters ● string manipulation functions ● comparing strings	2	2	**[1] 7.1, 7**.2, **[2]** 6A, **[4]** 12.1-12.3, 12.6 **[1]** 7.3, **[2]** 6A, **[4]** 8.2, **[4]** 12.7	*Laboratory work #8*
10	**Introduction to Functions:** ● Function Definition ● Custom functions, built-in functions ● Returning a Value, void functions ● Techniques of Passing Arguments	2	2	**[1]** 7.5, **[2]** 6A, **[4]** 19.1 **[1]** 7.5, **[2]** 6Б, **[4]** 8.3, **[4]** 19.1	*Laboratory work #9*
11	**Introduction to Pointers Pointers:** ● Operations on Pointers	2	2	**[2]** 6A, 6Б **[2]** 7A, **[4]** 9.1, 9.2,	*Laboratory work #10*

	● Passing Pointers to Functions ● Pointers and Memory Management				
12	**Struct, header files:** ● Structure Definition ● Syntax of structure ● Structure variable declaration ● Accessing members of a structure ● Structures within structures ● Passing structures to a function ● Headers, and their purpose ● Using standard library header files ● Writing your own header files	2	2	**[1]** 8.1, **[2]** 7А, **[4]** 13.1-13.3,13.6 **[1]** 8.2, **[2]** 7А, 7Б, **[4]** 13.4, 13.9, 14.3, 14.4	*Laboratory work #11*
13	**STL 1:** ● Vector ● Set ● Map ● Iterators	2	2	**[1]** 6.1, 6.2, **[2]** 7А, 7Б, **[4]** 8.1-8.3 **[1]** 6.2, **[2]** 8А, **[4]** 10.1, **[4]** 6.3, 8.4, 8.5	*Laboratory work #12*
14	**STL 1:** ● Queue ● Stack ● Multimap, multiset ● Next_permutation, sort	2	2	**[1]** 6.3, **[2]** 8А, **[4]** 10.2, **[4]** 9.1-9.3, 9.6 **[1]** 6.4-6.6**, [2]** 8Б**, [3]** 10.3.2, **[4]** 10.1, 10.2, 10.4, 10.6-10.8	*Laboratory work #13*
15	*Endterm*	2	2	**[2]** 8Б**, [3]** 10.3.2, **[4]** 11.1-11.4, 11.6 **[1]** 5.1-5.3, **[2]** 8Б**, [3]** 10.3.2	
16-17	Final Exam		2	Online contest format	

Laboratories: The preparation to the laboratories is provided in the form of solving of typical problems according to the lectures topics.