JSC «Kazakh-British Technical University»
Faculty of Information Technology

APPROVED BY
Dean of FIT
Suliev R. N. _____
«____»_____ 20__.

SYLLABUS

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

Personal Information about the Instructor	Time and place of classes		Contact information
	Classes	Office Hours	e-mail
Askar Akshabayev	According to the schedule	Room 279, will be appointed	a.akshabaev@kbtu.kz
Beisenbek Baisakov	According to the schedule	Room 272, will be appointed	b.baisakov@kbtu.kz
Bobur Mukhsimbayev	According to the schedule	Room 260, will be appointed	b.mukhsimbaev@kbtu.kz
Alimzhan Amanov	According to the schedule	Room 260, will be appointed	a.amanov@kbtu.kz
Zhasdauren Duisebekov	According to the schedule	Room 263, will be appointed	z.duisebekov@kbtu.kz

COURSE DURATION: 4 credits, 15 weeks

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 the principles of programming using C++ rather than attempting to give a 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
5. List of tutorials and portals for practical training
    a. https://informatics.mccme.ru/
    b. https://www.codewars.com
    c. https://www.coderbyte.com/
    d. https://codeforces.com/
    e. https://www.hackerrank.com/
    f. https://www.codecademy.com/learn/learn-c-plus-plus
    g. https://www.w3resource.com/cpp-exercises/basic/index.php
    h. https://www.programiz.com/cpp-programming/examples
    i. https://www.cprogramming.com/

Week	Class work	
	Topic	**Laboratory work**
1	**L1. Introduction to C++** ● Introduce Syllabus ● What is programming? ● Introduction to code structure ● Compiling and executing program ● Variables, declaration of variables ● Arithmetic operations ● Assign values ● Introduction to data types ● int, double, float ● char, string (type casting), concatenation ● bool (and, or, xor) ● Comments ● Math functions (sqrt, abs, sin, max, min, pow) ● Introduction to git	*Laboratory work #1*
2	**L2. Variable and Data Types** ● Introduction to numeric systems ● Logical Operators (and, or, xor, not) ● Logical Comparisons ● If else statement (nested if else statements) ● Math functions - pow, round, ceil, floor ● Introduction to Char, String ● Bit Manipulations ● Loop operators (for, while, do while)	*Laboratory work #2*

	● continue, break operators in loops ● freopen	
3	**L3. What is an array?** ● Types of Arrays ● Array declaration ● Accessing element of array ● Searching in Array ● 1D array samples ● String as array of chars	*Laboratory work #3*
4	**L4. Two-Dimensional Arrays?** ● Infinity loop, nested loops ● Initializing 2D arrays ● Accessing 2D array elements ● Examples for 2D array (matrix) ● Array sort, reverse	*Laboratory work #4.*
5	**L5. String** ● Initialize string (with constructor) ● size & length ● Copy string from one to other ● Comparing two string for equality ● String concatenation ● Accessing each element of the string ● Convert char to number (ASCII code) ● front, back, begin, end ● find, getline, substr, stoi ● erase, append, insert ● StringStream	*Laboratory work #5*
6	**L6. Functions** ● Built-in functions (sort, reverse, tolower, toupper, isalpha, isdigit, isalnum, ispunct, sqrt, max, min) ● What is a function? ● Function calling ● Function declaration and definition ● Function params ● Types of functions (Returning data, void) ● Local and global variables	*Laboratory work #6*
7- 8	**Midterm exam**	
9	**L9. What is a recursion?** ● Base case ● Stack overflow ● Examples: (factorial, power, fibonacci, min, max)	*Laboratory work #9*
10	**L10. STL**	*Laboratory work #10*

	• Algorithms, Containers, Functions, Iterators • Containers (vector, queue, stack, set, map) • What is a Vector? Manipulation on it using built-in functions • What is a Set? Manipulation on it using built-in functions • What is a Map? Manipulation on it using built-in functions	
11	**L11. STL (cont.)** • What is a Stack? Manipulation on it using built-in functions • What is a Queue? Manipulation on it using built-in functions • What is a Deque? Manipulation on it using built-in functions	*Laboratory work #11*

12	**L12. Library <algorithm>**
	• count_if (begin, end, function)
	• rotate (begin, middle, last)
	• fill (begin, end, val)
	• unique (begin, end)
	• for_each(begin, end, function)
	• generate(begin, end, gen_func)
	• lib <cstdlib>
	  ○ random value
	  ○ srand(time(0))
	• next_permutation, prev_permutation
		Laboratory work #12

13	**L13. Pointer and Struct**
	• What is a pointer?
	  ○ Declaring pointers
	  ○ Fill array with pointer
	  ○ Looping through array using pointers
	• What is struct?
	  ○ Constructor
	  ○ Header file
		Laboratory work #13

14-15	**End Term**	
16	**Final Exam**	

COURSE ASSESSMENT PARAMETERS

Type of activity	Final scores
Midterm	30%
Labs	0%
End Term (Project)	30%
Final exam	40%
Total	**100%**

Criteria for evaluation of students during semester:

	Assessment criteria	Weeks															Total scores
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1.	Midterm							*	*									30%
2.	Labs	*	*	*	*	*	*			*	*	*	*	*				0%
3.	Endterm														*	*		30%
3.	Final exam																*	40%
	Total																	**100%**

Academic Policy

KBTU standard academic policy is used.

- Cheating, duplication, falsification of data, plagiarism, and crib are not permitted under any circumstances!
- Attendance is mandatory.

  Attention. Missing 20% attendance to lessons, students will be taken from discipline with filling in F (Fail) grade.

  Students must participate fully in every class. While attendance is crucial, merely being in class does not constitute “participation”. Participation means reading the assigned materials, coming to class prepared to ask questions and engage in discussion.
- Students are expected to take an active role in learning.
- Written assignments (independent work) must be typewritten or written legibly and be handed in time specified. Late papers are not accepted!
- Students must arrive to class on time.
- Students are to take responsibility for making up any work missed.
- Make up tests in case of absence will not normally be allowed.
- Mobile phones must always be switched off in class.
- Students should always be appropriately dressed (in a formal/semi-formal style).
- Students should always show tolerance, consideration and mutual support towards other students.