

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

**APPROVED BY**

**Dean of FIT  
Suliyev R. N.**

« \_\_\_\_ » \_\_\_\_\_ **2021.**

**SYLLABUS**

**Discipline:** CSE212 Object-Oriented programming and Design

**Number of credits:** 3 (2/0/1)

**Term:** Fall-2021

**Instructor's full name:** Zhasdauren Duisebekov

Information about the Instructor	Time and place of classes		Contact information
	Lessons	Office Hours	e-mail
Duisebekov Zh.A. Senior-lecturer	Tuesday 09:00 – 11:50 Wednesday 09:00 – 11:50 MS Teams	Wednesday 15:00 – 17:00 MS Teams (by appointment)	<a href="mailto:z.duisebekov@kbtu.kz">z.duisebekov@kbtu.kz</a>

**Course duration:** 3 credits, 15 weeks (45 class hours)

**Course pre-requisites:** Programming Principles

**Course Objectives:**

Object Technology has been in development for over forty years. It is now embedded in such diverse areas as requirements engineering, software architecture, analysis, design, programming, testing, deployment and maintenance. The most widely used modern programming languages C++, Java and VB.Net all embrace an object oriented approach. This module examines the application of the object oriented paradigm to programming. The fundamental concepts of object oriented programming will be studied using the Java programming language. The main objective of the course is to develop an understanding of the principles underpinning object oriented programming and apply object-based approaches using Java programming language.

**Course Goals:**

During the course student will obtain the knowledge of fundamental principles of Object-Oriented programming. In particular, he/she will become familiar with the concepts of classes, objects, interfaces, inheritance, nested and abstract classes, polymorphism and data encapsulation. Moreover, by the end of the course he/she will master basic Java constructs – java collections (sets, lists, maps), exceptions, input / output, etc. The main goal to be achieved during this course is to be able to use a high-level object-oriented programming language as a problem-solving tool - including basic data structures and algorithms, object-oriented programming techniques, and software documentation.

**Literature:****Required:**

- Daniel Liang. *Introduction to Java Programming* (Comprehensive version, 10th edition)

**Supplementary:**

- Java API documentation: <https://docs.oracle.com/javase/7/docs/api/>

**COURSE CALENDAR**

Week	Class work					SIS (students independent study)		TSIS (teacher supervised independent study)	
	Topic	Lectures, hours	Labs, hours	Seminars, hours	Chapters for reading	Hours	Description	Hours	Description
1	<b>Introduction</b>  Java features. JVM. JRE. Java syntax. Java API. Variables. Programming fundamentals. Intro to OOP.	2	0	1	Ch.1-8	1	Ch.1-8 Exercises	3	
2	<b>Objects and Classes</b>  Class vs objects. UML. Constructors. References.	2	0	1	Ch.9	1	Ch.9 Exercises	3	
3	<b>Objects and Classes</b>  Data fields and methods. Primitive vs Reference types. Java library classes.	2	0	1	Ch.9	1	Ch.9 Exercises	3	
4	<b>Objects and Classes</b>  Instance vs static fields. Encapsulation. Array of objects. Scopes. <i>this</i> keyword.	2	0	1	Ch.9	1	Ch.9 Exercises	3	
5	<b>Object-Oriented Thinking</b>	2	0	1	Ch.10	1	Ch.10 Exercises	3	

	Class abstraction. Procedural vs Object-oriented paradigms. Class relationships. Object-oriented design.								
6	<b>Object-Oriented Thinking</b> Wrapper classes. String classes.	2	0	1	Ch.10	1	Ch.10 Exercises	3	
7	<b>Inheritance and Polymorphism</b> Superclasses and subclasses. The <i>super</i> keyword. Overloading vs overriding.	2	0	1	Ch.11	1	Ch.11 Exercises	3	
8	<b>Inheritance and Polymorphism</b> Object class and its methods. Polymorphism and dynamic binding. ArrayList. <i>protected</i> and <i>final</i> keywords.	2	0	1	Ch.11	1	Ch.11 Exercises	3	
9	<b>Midterm Exam</b>	2	0	1		1		3	
10	<b>Exception Handling and Text I/O</b> Advantages. Error vs Exception. Throw, try, catch. File, PrintWriter, and Scanner classes.	2	0	1	Ch.12	1	Ch.12 Exercises	3	Assignment of Student Project
11	<b>Abstract Classes and Interfaces</b> Generalization. Defining interfaces and classes that implement interfaces. Comparable. Cloneable.	2	0	1	Ch.13	1	Ch.13 Exercises	3	
12	<b>Generics</b> Benefits. Generic classes and interfaces. Generic methods. Raw types. Wildcards. Limitations.	2	0	1	Ch.19	1	Ch.19 Exercises	3	Checkpoint
13	<b>Lists, Stacks, Queues, and Priority Queues</b> Collection interface. Iterator. ArrayList vs LinkedList. Comparable vs Comparator. Collections class.	2	0	1	Ch.20	1	Ch.20 Exercises	3	
14	<b>Sets and Maps</b> Set classes. Sets vs Lists. Collection	2	0	1	Ch.21	1	Ch.21 Exercises	3	Project Defense

	vs Map.								
15	<b>Implementing Lists, Stacks, Queues, and Priority Queues</b>  Designing an array list and a linked list. Implementing a stack and a queue.	2	0	1	Ch.24	1	Ch.24 Exercises	3	Project Defense
	<b>Final Exam</b>						<b>In written form</b>		
	<b>Total</b>	<b>30</b>	<b>0</b>	<b>15</b>			<b>15</b>		<b>45</b>

### COURSE ASSESSMENT PARAMETERS

Type of activity	Final scores
Practice activities	10%
Project	20%
Midterm exam	30%
Final exam	40%
<b>Total</b>	<b>100%</b>

### 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	16-17	
1.	Practice activities						*	*	*	*	*	*	*	*	*	*	*	10%
2.	Project														*	*		20%
3.	Midterm exam								*									30%
4.	Final exam																*	40%
	<b>Total</b>																	<b>100%</b>

### 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, student 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.