JSC «Kazakh-British Technical University»
School of IT and Engineering

APPROVED BY
Dean of SITE
Alibek Bissembayev

_____
«____» _____ 2022

SYLLABUS

Discipline: CSE212 Object-Oriented programming and Design
Number of credits: 3 (2/0/1)
Term: Fall 2022
Instuctor's full name:

Personal Information about the Instructor	Time and place of classes		Contact information	
	Lessons	Office Hours	Telegram:	e-mail
Shamoi Pakizar, Professor, Ph.D.	According to the schedule	According to the schedule	pakita883	p.shamoi@kbtu.kz

COURSE DURATION: 3 credits, 15 weeks (45 class hours)

GENERAL COURSE AIMS:

(1) To obtain the knowledge of fundamental and advanced principles of Object- Oriented programming
(2) To be able to use a high-level object-oriented programming language as a problem-solving tool

COURSE DESCRIPTION

The object technology is the driving force in the programming industry. It is now embedded in such diverse areas as requirements engineering, software architecture, analysis, design, programming, testing, deployment and maintenance. The fundamental concepts of object-oriented programming will be studied using the Java programming language. You will discover how to create flexible and reusable software, by applying object-oriented design principles and guidelines. And, you will be able to communicate these designs in a visual notation known as Unified Modelling Language (UML). OOP is crucial to all modern software development—including Applications development, Web development, and Mobile Applications Development. Moreover, OOP defines most modern server-side scripting languages.

COURSE OBJECTIVES

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 OUTCOMES

Students will be exposed to basic and advanced oop concepts (Encapsulation, Inheritance, Polymorphism, Abstraction, Interfaces, Design Patterns, UML etc.). These allows to achieve the following outcomes which are critical success factors for any project in IT industry: Ease of software design, Productivity, Easy testing, Debugging, and Maintenance, Code reuse, Less development time, and more accurate coding.

COURSE POST REQUISITES

Knowledge and skills obtained during the study of this course are used in the following courses: Web development, Mobile Applications Development, Software Emgineering.

LITERATURE:

Required:

1. Pakizar Shamoi. Object-oriented Programming and Design. KBTU, 2013

Supplementary:

2. Java API documentation: https://docs.oracle.com/javase/7/docs/api/index.html?overview-summary.html
3. Effective Java, Joshua Bloch.
4. Object-oriented Software Engineering: Practical Software Development Using UML and Java (Book by Robert Laganière and Timothy C. Lethbridge)
5. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Publishing Company
6. Ivor Horton. Beginning Java™ JDK™ 5 Edition. Wiley Publishing, 2005.
7. Fowler M., UML Distilled, Addison-Wesley (3rdEd).
8. Weisfeld M., The Object-Oriented Thought Process, Addison-Wesley Professional (4th Ed.).

COURSE ASSESSMENT CRITERIA

Assessment occurs continuously throughout the course. The evaluation will be based on the levels of (maximums in %):

COURSE ASSESSMENT PARAMETERS*

Type of activity	Final scores
Laboratory works	14%
Quizzes	2%
SIS	20%
Project	10%
Midterm, Endterm exams	14%
Final exam	40%
Total	**100%**

*Can be changed with prior announcement

TASKS for students independent study (SIS)

Week	SIS	Cost (in points)
4	SIS 1 (Objects, Classes, Inheritance)	5
6	SIS 2 (Interfaces, Polymorhism)	5
9	SIS 3 (UML model)	5
12	SIS 4	5
	Total	20

TASKS (laboratory works/practice)

Week	TSIS	Cost (in points)
2	Lab1	2
3	Lab2	3
5	Lab3	4
7	Lab4	5
	Total	14

COURSE CALENDAR

Week	Class work					SIS (students independent study)		TSIS (teacher supervised independent study)	
	Topic	Seminars, hours	Lab, hours	Lectures, hours	Chapters for reading	Hours	Description	Hours	Description
1	**Lecture #1. Object Orientation as a New Paradigm . Introduction to Java programming language.** Java features. JVM. JRE. Java syntax. Java Api. Variables. Data types used in Java. Control statements. OO Approach. Genealogy of object oriented languages. Comparison to procedural programming.	1	0	2	Ch [1],[2]	1	Lab #1,	3	

2	**Lecture #2. Fundamentals of Objects and Classes.** Class members and instance members. Access control. Creating objects. Methods. Constructors. Initialization blocks. Methods overloading.	1	0	2	Ch [3]	1	Lab #1	3	
3	**Lecture #3. Fundamentals of Objects and Classes.** Packages. Naming conventions. Class importation. Class design. Java API and Core Java classes	1	0	2	Ch [3]	1	Lab #2, SIS1	3	
4	**Lecture #4. Inheritance, Polymorphism and Abstract classes.** Relationships among classes. Extending existing classes. Inheritance and class hierarchy.	1	0	2	Ch [4]	1	Lab #2, SIS1	3	TSIS 1
5	**Lecture #5. Inheritance, Polymorphism and Abstract classes.** Methods overloading and overriding . Subclass and Superclass. Type Conversion. Polymorhism	1	0	2	Ch [4]	1	Lab #3	3	
6	**Lecture #6. Inheritance, Polymorphism and Abstract classes.** Abstract classes and methods. Special variables *this* and *super*. Design hints for inheritance.	1	0	2	Ch [4]	1	Lab #3	3	
7	**Lecture #7. Interfaces** Interface members. Extending interfaces. Object cloning. Interfaces and abstract classes.	1	0	2	Ch [5]	1	SIS2	3	
8	**Lecture #8. Interfaces** Marker interfaces. Cloneable and Comparable interfaces. Nested classes. Difference between Interfaces and Abstract classes.	1	0	2	Ch[5]	1	Lab#4	3	TSIS2
9	**Lecture #9. Collections & Data Structures**	1	0	2	Ch [7]	1	Lab#5	3	

	HashSet. TreeSet. HashMap. LinkedList. Lists. Vector. Stack. Iteration. Sorting a collection. Implementing user-defined collections. *Viewing and analyzing Collections hierarchy, demonstrating OOP principles realization.*								
10	**Lecture #10. Files and streams. Exceptions** Streams, Readers and Writers. PrintWriter. Scanner. RandomAccessFile, Buffered Streams.InputStream/OutputStream. Reader/Writer. *Viewing and analyzing Exceptions hierarchy, demonstrating OOP principles realization.* Checked & Unchecked exceptions. Claiming Exceptions. Throwing Exceptions . Catching Exceptions Handling the exceptions. try-catch block, finally clause. *Viewing and analyzing Streams hierarchy, demonstrating OOP principles realization.*	1	0	2	Ch [8],[9]	1	SIS3, Lab #5, #6	3	Assignments of Student Project
11	**Lecture #11. Threads** Creating and executing threads. Managing threads. Thread synchronization. Life-cycle of a thread. Thread priorities. Thread Groups. *Viewing and analyzing Threads hierarchy, demonstrating OOP principles realization.*	1	0	2		1	SIS #3	3	
12	**Lecture #12. UML diagrams and Software Documentation** Use Case, Sequence , Class diagrams. Realization, Dependency, Aggregation,	1	0	2	[1] Ch. 23 [2] Ch.	1	SIS #4	3	Checkpoint

	Composition. Acquaintance with TopCoder UML tool. Software Documentation. Javadoc. Coupling and cohesion.				16				
13	**Lecture #13. Design Patterns. Overview.** Singleton pattern, abstract factory pattern, prototype pattern, Decorator, etc.	1	0	2	Slides	1	SIS #4	3	
14	**Lecture #14. Design Patterns (Advanced)**	1	0	2	Slides	1	Project	3	Project Defense
15	**Lecture #15 Recent Advances in Component Software** Object-oriented and functional programming. Scala. Twitter example.	1	0	2	Slides All chapters	1		3	
	Final Exam					**In written form**			
	Total	**0**	**30**	**30**		**15**		**45**	

Examination Format: Project defense in oral form or exam in written form

Class sessions – will be a mixture of information, discussion and practical application of skills.

Practice – will reinforst the students knowledge by practical appliance of lectured materials.

In-class assessment – will prepare students for their mid-term and final assessment and identify the competence level they have achieved on a related subject matter, the aim being to diagnose potential discrepancies in students' understanding and performance in order to make specific adjustments to the course content and procedures and/or to assign additional assignments to certain individuals or the whole group.

Home assignments – will consolidate the concepts and materials taken during in-class activities, help students to expand the content through diverse background resources and/or practise certain skill areas; they will also develop the students' ability to work individually in exploring and examining related issues.

SIS (Student Independent Study) – comprises group Project to be done by students on the independent basis. Students are supposed to use knowledge and skills acquired in class to do the project. Assistance and advice will be provided by teachers during office hours**.**

TSIS (Teacher Supervised Student Independent Study) – student self-made project.

End-term test – a diagnostic test used to identify the students' progress, their strengths and weaknesses, intended to force student to prepare for Final Exam. It includes computer based test.

Final examination – 1) an attainment test designed to identify how successful the students have been achieving objectives.

Grading policy:

Intermediate attestations (on 7th and 15th week) join topics of all lectures, practice, laboratories, SIS, TSIS and materials for reading discussed to the time of attestation. Maximum number of points within attendance, activity, SIS, TSIS and laboratories for each attestation is 30 points.

Final exam joins and generalizes all course materials, is conducted in the complex form with quiz and problem. Final exam duration is 100 min. Maximum number of points is 40. At the end of the semester you receive overall total grade (summarized index of your work during semester) according to conventional KBTU grade scale.

Attention!

1) If student missed without plausible reason more than 20% of lessons student receives «F (Fail)» grade;
2) If for two attestations student receives 29 or less points, this student is not accepted to final exam and for all course he (she) receives «F (Fail)» grade;
3) If student receives on final exam 19 or less points, then independently on how many points he (she) received for two attestations, in whole he (she) receives «F (Fail)» grade;

In the case of missing or being late for final exam without plausible reason, independently on how many points he (she) received for two attestations, in whole he (she) receives «F (Fail)» grade.

Students are encouraged to

- consult the teacher on any issues related to the course;
- make up within a week's time for the works undone for a valid reason without any grade deductions;
- make any proposals on improvement of the academic process;
- track down their continuous rating throughout the semester.

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.