BAKI ALİ NEFT MƏKTƏBİ
BAKU HIGHER OIL SCHOOL

The Ministry of Education of the Azerbaijan Republic
The State Oil Company of the Azerbaijan Republic
Baku Higher Oil School

Information Technology Department

Object Oriented Programming

Information Security Bachelor's degree program

Courses Syllabus

Spring, 2025

Instructor	:	Elvin Taghizade
Course code	:	IS 218
Course credit	:	6
Office	:	Campus Aypara
Prerequisites	:	Java programming
Language of instruction	:	English
Email	:	*elvin.taghizade@bhos.edu.az*
Schedule	:	

- IT 23.1
- Frequency: Twice a week, with two sessions each time.
- Days and Times:
  - Wednesday: 13:00–15:00, room 402
  - Saturday: 14:00–16:00, room 211

Description about course

Object-oriented programming (OOP) is a programming language model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior. Object-oriented programming is based on the imperative programming paradigm, which uses statements to change a program's state. It focuses on describing how a program should operate. This stands in contrast to declarative programming, which focuses on what the computer program should accomplish, without specifying how. The course includes basics of OOP conceptions of inheritance, abstraction, view onto the similar object. Second part of this course also includes an object oriented design pattern approach in order to solve the software developing problems based on the real world examples.

Course objective section

The course aims to present modern OOP approaches to many software developing problems to implement them in real life. The OOP approach teaches techniques to solve developing problems in most efficient (from the organization point of view) ways. After this course students will be able to write large scalable software, organize the code in packages and build it in an effective way.

Learning outcomes section

- Learning the basic conception of OOP paradigm
- Create flexible software with abstract separation, extensibility and variability
- Problem solving with help of structural, behavioral,
- Learn to create and use Linear & Non-Linear Data Structures, Trees and Graphs

Assessment methods

The exams are done using computers. Some parts of exams are written examinations. All questions must be answered.

Grading

Exam	Weight	Date	Exam minutes
Final	50%	TBA	180
Project	30%	during the semester	deadline
Quiz	10%	once during semester	60
Attendance	10%	at the end of semester	participation/lessons

NOTE: Total number of labs is ten. Each lab is graded separately. At the end of the term the average mark of the labs is calculated.

RESIT grading (in case of fail Final exam**):**

Weight: **70%** Date: TBA (to be announced) Exam duration: 180 minutes

Total score: Resit Exam score (70%)+project score (30%)

Area grading scale

A	91-100
B	81-90
C	71-80
D	61-70
F	≤ 60

Rules

Exams

In order to be excused from the exam, the student must contact the dean and the instructor before the exam. Excuse will not be granted for social activities such as trips, cruises and sporting events (unless you are participating). The exams will all be cumulative. Most of the questions on each exam will be taken from the chapters covered since the last exam.

But some will come from the earlier chapters. In general, the coverage will reflect the amount of the time spent in class on the different chapters.

Withdrawal (pass / fail)

This course strictly follows the grading policy of the Process Automation Engineering Department. Thus, a student is normally expected to achieve a total mark (pre exam score + exam score) of at least 61 to pass. In this case of failure, he/she will be referred to pass Resit or required to repeat the course the following term or year in case of failed Resit.

Late policy

Late assignment submissions won't be accepted for grading. The grade for this assignment will be zero.

Teaching resources

1. Thinking in Java 4th edition by Bruce Eckel
2. Effective Java (3rd Edition) by Joshua Bloch
3. Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson
4. Head First Java (2nd Edition) by Kathy Sierra and Bert Bates

For class presentations and discussions, the student should utilize journal and internet materials. Moreover, the course does not limit the use of learning materials available at BHOS library.

Attendance

The students are required to attend all classes as a part of their studies and those having legitimate reasons for absence (illness, family bereavement, etc.) are required to inform the instructor.

Professionalism and Participation

1. Attend class regularly, arrive on time, leave only when dismissed
2. Attend class with all materials required, be prepared to listen and work
3. Be well prepared for class, read all required materials, and complete all necessary preparation
4. Be attentive in class, take notes, contribute to discussion and ask intelligent questions
5. Demonstrate professional and respectful interpersonal relationships with peers and instructor: ATTITUDE COUNTS, AND whining is unacceptable
6. Take responsibility for your actions, and your results

Plagiarism

Honesty requires that any ideas or material taken from another source for written, visual, or oral use must be fully acknowledged. Offering the work of someone else as one's own is plagiarism. The language or ideas thus taken from another may range from isolated formulas, images, sentences or paragraphs to entire articles copied from books, periodicals, speeches, or the writings and creations of other students. The offering of materials assembled or collected by others in the form of projects or collections without acknowledgment also is considered plagiarism. Any student who fails to give credit for ideas or materials taken from another course is guilty of plagiarism.

Week	Topics	Textbook/Assignments
1	**Introduction to Programming Paradigms** - Fundamentals of Programming - Intro to Object Oriented Programming - Intro to Functional Programming	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
2	**Basics of OOP** - Object and class, UML diagrams - Constructors, object initialization - Types of variables - Instance variables - Local variables - Static (global) variables - Static vs non-static methods & variables - References/Garbage Collection - Getters and setters	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
3	**Classes and Objects in Java** - Creating classes and Objects - Adding attributes - Creating methods - Creating attributes - Object Initialization - Organizing modules and Packages - Access Control	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
4	**Encapsulation** - Access modifiers - Getters-setters - Quiz & Practice	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
5	**Inheritance** - Inheritance and multiple inheritance - Diamond problem - IS-A relationship - HAS-A relationship - Object class - toString(), equals(), hashcode() - Abstract classes and methods	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates

Week	Topics	Textbook/Assignments
6	**Polymorphism** - Polymorphism - Abstraction - Quiz & Practice - this & super & instanceof - @Override - Compile-time (overloading) vs runtime (overriding) polymorphism	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
7	**Inheritance & Polymorphism in action** - Abstract classes - Interfaces - Abstract classes vs interfaces in depth - Functional & Marker Interfaces - Quiz & Practice	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
8	**Enumeration & Immutability** - Enumeration - Immutability - Final class - Final method - Final fields, parameters - recursive immutability - Var keyword - Casting (upcating, downcasting) - Boxing and unboxing. Autoboxing	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
9	**Exception Handling & Date/Time lib** - Exception hierarchy - Error vs Exception - Checked and unchecked exceptions - Try-catch - Multiple catch and union catch - Swallowing exceptions - Custom Exceptions - throw vs throws - LocalDate & LocalTime & LocalDateTime	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates

Week	Topics	Textbook/Assignments
10	**Object Oriented Design principles** - DRY - KISS - SOLID - YAGNI	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
11	**OOP Design patterns: Introduction to GOF design patterns** - Design patterns types - Singleton pattern - Double-checked Singleton pattern	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
12	**OOP Design patterns: Creational patterns** - Prototype - Factory pattern - Factory method - Abstract Factory - Builder pattern	[1] Thinking in Java 4th edition by Bruce Eckel [2] Effective Java (3rd Edition) by Joshua Bloch [3]Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson [4] Head First Java (2nd Edition) by Kathy Sierra and Bert Bates
	Final Exam	

Instructor of the course **Elvin Taghizade**

Head of the department ______________