Department of Software Engineering

College of Engineering

University of Salahaddin-Erbil

Subject: Advanced *Object-Oriented Programming*

Course Book

Kanar Shukr Muhamad

Academic Year: 2023/2024

Course Book

1. Course name	**Object Oriented Programming**
2. Lecturer in charge	**Kanar Shukr Muhamad**
3. Department/ College	**Software Engineering\Engineering**
4. Contact	**e-mail: kanar.muhamad@su.edu.krd**
5. Time (in hours) per week	**Theory: 3**
6. Office hours	**Sunday 9:00 am – 11:00 am**
7. Course code	
8. Teacher's academic profile	Kanar Shukr Muhamad Lecturer at Software and Informatics Engineering Department College of Engineering – Salahaddine University Hawler – Kurdistan **Current**: Advanced Object-Oriented Programming (HD) Data Structures and Algorithms (2nd Year) Numerical Analysis (2nd Year) **Past:** Programming Algorithms (1st Year), Programming Principles (1st Year), Procedural Programming (1st year), Digital Design (1st year), Data Structures and Algorithms (2nd year), Object Oriented Programming (2nd year) , Numerical Analysis and Probability(2nd year), Engineering Analysis (3rd year) and Data Security (4th year). **Education:** BSc of Software Engineering – college of Engineering – Salahaddine University (2005) MSc of Software Engineering – college of Engineering – Salahaddine University (2009) https://academics.su.edu.krd/kanar.muhamad/
9. Keywords	OOP, Polymorphism, Inheritance, Interface, Encapsulation

10. Course overview:

This course is a mandatory requirement for the HD in Software Engineering. It provides an introduction to the concepts and methodology of Object-Oriented Programming with Java as an illustration language. Students' previous exposure to and skills in procedural programming is assumed (although a review of the necessary basic programming notions is included).

11. Course objective:

- Understand major concepts of object-oriented programming
- Knowledge and skills in OO design and program development
- Experience in Java programming and program development within an integrated development environment
- Certain skills in Internet and windows programming and using graphical user interface.
- It also introduces the file types and their use in applications and programs.

12. Student's obligation

- Commitment to the Class and schedule.
- Cheating is prohibited and the student will be awarded zero.
- In case there is an assignment, deadline date must be aced. No delay will be accepted.

13. Forms of teaching

A compilation of lecture slides is provided in the class and will form the core of the course. However, additional materials may offer during the presentation of the course such as software, seminars, tutorials and websites.

14. Assessment scheme

The following grade system is used for the evaluation of the module exam:

The module exam is based on the summation of two categories of evaluations:

First: (50%) of the mark is based on the academic semester effort which includes

- Mid-term Exams- Quizzes- Activities- Reports- Seminars.

Second: (50%) of the mark is based on final examination that is comprehensive for the whole of the study materials reviewed during the academic semester.

Note:

There will be randomly quizzes. Each quiz will be given at the beginning of the class period and covers materials covered in the previous lectures.

15. Student learning outcome:

Student finishes this semester successfully should be able to put object-oriented methodologies under practice.

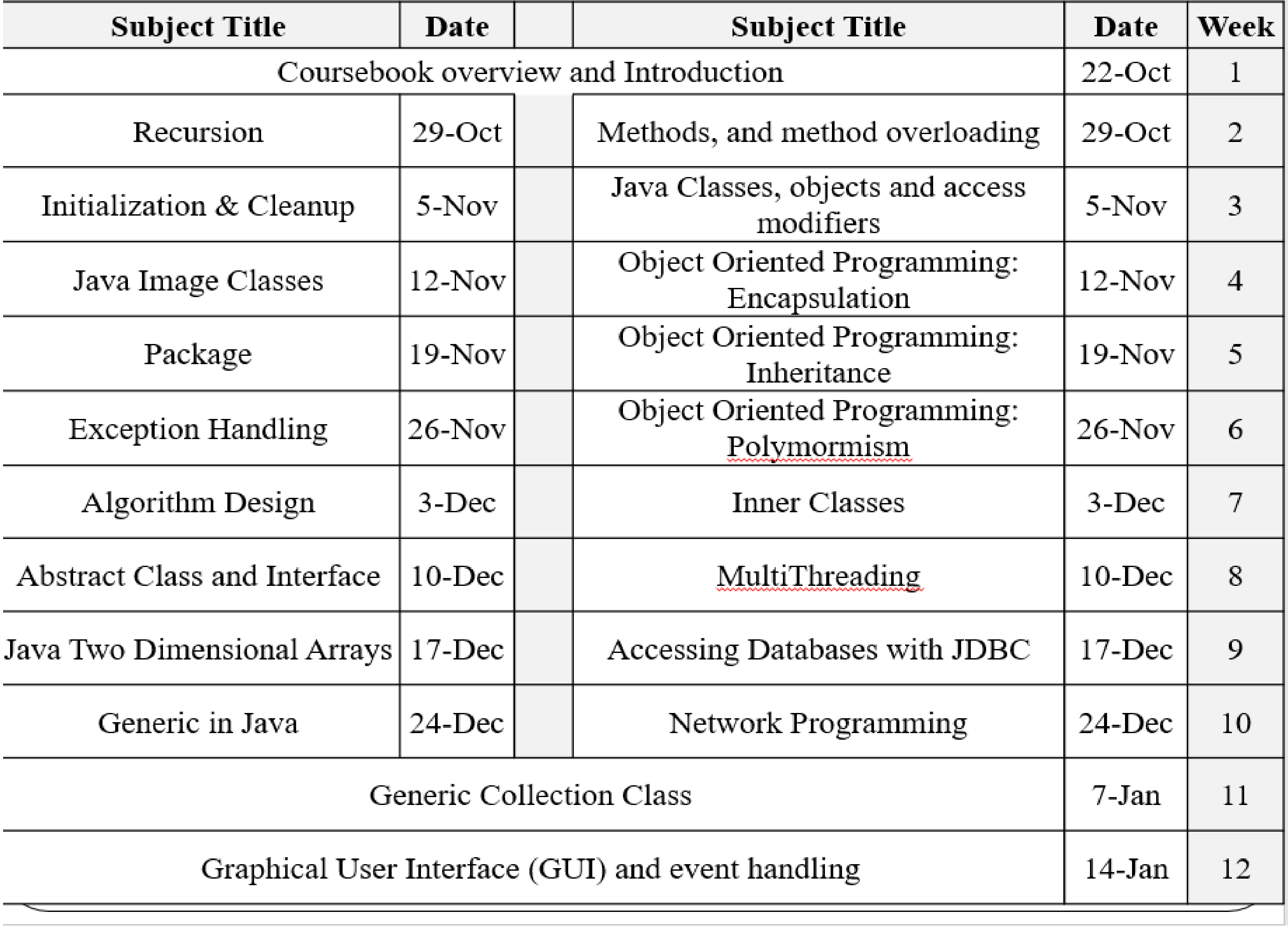
At the end of this course, students should be able to:

- write complex and large OO programs in Java
- demonstrate a deep understanding of OO concepts
- write sophisticated GUI programs
- utilize advanced features of Java in their programs

16. Course Reading List and References:

- Paul Deitel, Harvey Deitel., " Java How to Program ", 10th Edition, ISBN 978-0132575669.
- Ralph Morelli, Ralph Walde, “Java, Java, Java™: Object Oriented Problem Solving”, 3rd Edition, June 25, 2017
- Herbert Schildt , “Java™ : The Complete Reference”, 7th Edition, ISBN: 978-0-07-163177-8, 2007.
- B. Eckel , “Thinking in Java”, 4th ed., Prentice Hall, 2006

17. The Topics: **Lecturer's name:** kanar Sh. Muhamad



Subject Title	Date		Subject Title	Date	Week
Coursebook overview and Introduction				22-Oct	1
Recursion	29-Oct		Methods, and method overloading	29-Oct	2
Initialization & Cleanup	5-Nov		Java Classes, objects and access modifiers	5-Nov	3
Java Image Classes	12-Nov		Object Oriented Programming: Encapsulation	12-Nov	4
Package	19-Nov		Object Oriented Programming: Inheritance	19-Nov	5
Exception Handling	26-Nov		Object Oriented Programming: Polymormism	26-Nov	6
Algorithm Design	3-Dec		Inner Classes	3-Dec	7
Abstract Class and Interface	10-Dec		MultiThreading	10-Dec	8
Java Two Dimensional Arrays	17-Dec		Accessing Databases with JDBC	17-Dec	9
Generic in Java	24-Dec		Network Programming	24-Dec	10
Generic Collection Class				7-Jan	11
Graphical User Interface (GUI) and event handling				14-Jan	12

18. Examination:

Example 1/ Fill in blanks with the correct answer

1- When each object of a class maintains its own copy of an attribute, the field that represents the attribute is also known as a(n) **instance field.**

2- Classes from which objects can't be instantiated are called **abstract** classes.

3- **Polymorphism** involves using a superclass variable to invoke methods on superclass and subclass objects, enabling you to "program in the general."

4- Any class that subclasses the Thread class should provide override the **run()** method.

Example 2/ State whether each of the following is true or false. If false, explain why?

1- When events occur on an event source it sends a notification of that event to all the listener objects.

False, it sends a notification of that event to all the listener objects that are registered for that event.

2- A server waits at a port for connections from a client.

true

Example 3/ Try to answer all in brief.

1- Just as a program starts running, do any objects exist?

No. Objects are created one by one as the program runs.

2- What is the final keyword? and where it's used explain it briefly.

final: final keyword can be used for class, method and variables.

A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods.

A final method can't be overridden.

A final variable can't change from its initialized value.

3- Consider the following class definitions and determine the output that would be generated by the code segment.

```
     public class A {  public void method() {System.out.println("A");}}

public class B extends A { public void method(){System.out.println("B");}}
```

For the class B defined, modify its method() so that it invokes A's version of method() before printing out B.

void method () {super.method();System.out.println("B");}

Example 4/ Write a very simple class called **Rectangular** that defines two instance variables: **width** and **height**. **Initialize objects** using (parameterized and default constructors), create a method for computing a rectangular area, and a method named **equals** to check whether the two objects of type **Rectangular** are equal or not. Then write the **main** class, create objects of type **Rectangular** class and check the above methods.

Solution:

```
class Rectangular
{
     private double width;
     private double height;
//parameterized and default constructors
     public Rectangular(double w, double h)
     {
          width=w;
          height=h;
     }
     public Rectangular()
     {
          this(1.0,1.0);
     }
     // compute and return area
     public double area()
     {
          return width * height;
     }
     public boolean equals(Rectangular ob2)
     {
          return(width == ob2.width && height == ob2.height);
     }
}
```

```
public class OOPLectExample
{
     public static void main(String str[])
     {
          Rectangular b=new Rectangular();
          System.out.println(b.area());
          Rectangular b2=new Rectangular(2,3);
          System.out.println(b2.area());

          if(b.equals(b2))
                System.out.println("Yes");
          else System.out.println("NO");
     }
}
```

19. Extra notes: