T.C. ISTANBUL AREL UNIVERSITY						
FACULTY OF ENGINEERING						
DEPARTMENT OF COMPUTER ENGINEERING						
Academic	**Exam**			**Course**		
Year	**Type**	**Date**	**Time**	**Code**	**Name**	**Instructor**
Fall 2025	Midterm	20.11.2025	09:00-10:30	LCEN203	Object Oriented Programming	Tuğberk Kocatekin

INSTRUCTIONS

1. Cell phone use is prohibited.
2. Leaving the classroom within the first 15 minutes of the exam is prohibited.
3. Leaving the classroom during the last 5 minutes of the exam is prohibited.
4. You can enter the exam no later than 15 minutes after it starts.
5. There must be at least two students in the classroom until the end of the exam.
6. Students whose names are not on the exam list are prohibited from taking the exam.

Question:	1	2	3	4	5	6	7	Total
Points:	20	10	15	10	15	15	15	100
Score:								

Name: _____ ID: _____

Signature: _____

Questions

1. (20 points) Please answer the multiple choice questions below. There is only one right answer. Each question is worth 2 points.

   (a) Which of the following is TRUE about constructors in Java?
   - A. They can have a return type as long as it is void.
   - B. They must always be manually written; Java never provides one automatically.
   - C. They run automatically when an object is created.
   - D. They cannot take parameters.

   (b) What happens if you define a parameterized constructor but no empty constructor?
   - A. Java automatically generates both constructors.
   - B. Java generates only an empty constructor.
   - C. Java generates only a parameterized constructor.
   - D. Java does not generate an empty constructor.

   (c) Which statement best describes encapsulation?
   - A. Making all variables public for easier access.
   - B. Hiding data by making fields private and exposing controlled access via methods.
   - C. Allowing objects to modify each other's private fields directly.
   - D. Writing classes without any getter or setter methods.

   (d) Which of the following is TRUE about static variables?
   - A. Each object has its own independent copy.
   - B. They belong to the class and are shared among all objects.
   - C. They can only be accessed through an object instance.
   - D. They are automatically private.

   (e) Which one correctly describes an inheritance relationship?
   - A. A Truck is-a GPSModule.
   - B. A Teacher has-a Course.
   - C. A Manager is-a Employee.
   - D. A Student has-a GraduateStudent.

   (f) Which rule must be followed when overriding a method?
   - A. The method must have a different name.
   - B. The return type must always change.
   - C. The parameter list must remain the same.
   - D. The method must be declared static.

   (g) Which of the following is TRUE about interface fields?
   - A. They can be private instance variables.
   - B. They are implicitly public, static, and final.
   - C. They can be modified by implementing classes.
   - D. They cannot be accessed without creating an object of the interface.

   (h) Which statement is correct about abstract classes?
   - A. They cannot contain concrete (fully implemented) methods.
   - B. They cannot have constructors.
   - C. They cannot be extended by other abstract classes.
   - D. They can contain both abstract and non-abstract methods.

   (i) Which scenario correctly represents composition (has-a relationship)?
   - A. Car extends Engine
   - B. Employee extends Department
   - C. Department contains a list of Employee objects
   - D. GPSModule extends Truck

   (j) Which Java class is best suited for reading a file line-by-line efficiently?
   - A. Files.readAllLines
   - B. Scanner
   - C. BufferedReader
   - D. FileWriter

2. (10 points) Please give short and definitive answers to questions below. Each question is worth **1 points**.

   (a) A variable shared by all objects of a class is declared using the keyword: _____

   (b) Hiding internal data by making fields private is known as: _____

   (c) When a class contains another class as a field, this is known as the _____ relationship

   (d) In Java, interface fields are implicitly public, static, and _____

   (e) A class that provides partial implementation and must be extended is known as _____ class.

   (f) A subclass that does not implement all abstract methods from its parent must itself be declared _____

   (g) The principle recommending *prefer composition over inheritance* aims to reduce _____ coupling.

   (h) A subclass that overrides a method but still wants to call the parent version must prefix the call with _____

   (i) When a method in Java has the same name and same parameters in both parent and child classes, this is known as method _____.

   (j) Who is the creator of Java language? _____

3. (15 points) You are given three interfaces: **A**, **B** and **C**. Interface C extends both A and B. Class **D** implements interface **C**. Answer the questions below.

   (a) What is the relationship between interfaces C and A, B? (3p)

   (b) Fill in the blanks and write code for D, such that it is **not** an abstract class. (12p)

```
interface A {
  void methodA();
}

interface B {
  void methodB();
}

interface C extends A, B {
  void methodC();
}
class D _____________ C {

  //Write the code in the space on the right
}
```

4. (10 points) Below is an **interface** and a **class** which will use it. Fill in the blanks to complete.

   Interface:

```
interface Drawable {

    _______________________________();      (5p)
}
```

   Circle class:

```
public class Circle  ___________________________________ {    (5p)

    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

 @Override
    public void draw() {
        System.out.println("Drawing circle");
    }
}
```

5. (15 points) Here below you see two classes: Course and Student. Fill in the blanks.

Course class:

```
public class Course {

 _____________ String courseName; (1p)

 _____________ int credits; (1p)

   public Course(_____________________________, _____________________________) { 2p)

     this.courseName = tr;

     this.credits = cr;
 }

   public _____________ getCourseName() { (1p)

      return ____________________; (1p)
 }
}
```

Student class:

```
public class Student {

    _________________________________ name; (2p)

    _________________________________ course; (3p)


   public Student(String name, _____________________________) {  (2p)

       this.name = name;

       ________________________________________________; (2p)
 }
}
```

6. (15 points) Below there are two classes: **Person** and **Employee**. As we know, Employee *is-a* Person. Fill in the blanks so that code compiles.

Student class:

```
public class Person {

    ___________________ name;         (2p)

    ___________________ age;          (2p)

    public Person(String name, int age) {
```

```
        this.name = name;
        this.age = age;
    }
}
```

Employee class:

```
public class Employee ____________________________________ {    (5p)

    private double salary;

    public Employee(String name, int age, double salary) {

        _________________________________;            (5p)

        this.salary = ________________; (1p)
    }
}
```

7. (15 points) Look at the code below, what is the output of this code? It will give you **5 lines**. Write them in the empty space on the right in order.
Hint: Do not forget that every subclass runs the super() automatically.

```
class X {
  int x = 5;
  X() {
      System.out.println("i am x");
  }
  void print() {
    System.out.println("printing x");
  }
    }

class Y extends X {
  Y() {
    System.out.println("i am y");
  }

  @Override
  void print(){
    System.out.println("i overrid x");
    } }

  public static void main(String[] args) {
   X x = new X();
   x.print();

   X y = new Y();
   y.print();
  }
```