Homework 8

CSC-121, Fall 2025

Submission Instructions: Create a single text file for your answers (Use .txt or .pdf) with your name and ID number at the top. Include code written for any programs, and type-written answers to non-coding questions. Submit the file on Canvas to complete the assignment. You may work in groups or use genAI help, but written answers must be in your own words.

Q0. CodePath Mid-Semester Survey

How do you feel that the semester has been going so far? Please let us know in the mid-semester survey. This will help the teaching staff by providing feedback on what we could do to make the course better.

Survey Link: https://www.surveymonkey.com/r/cir-fall25-mc

Q1. Multiple Choice & True/False

Select only the best answer for each question. There is only one correct answer. You do not need to explain your choice.

1. A class is analogous to a(n) __________.
   a. house
   b. blueprint
   c. drafting table
   d. architect

2. An object is a(n) __________.
   a. blueprint
   b. primitive data type
   c. variable
   d. instance of a class

3. This is automatically provided for a class if you do not write one yourself.
   a. accessor method
   b. default instance
   c. default constructor
   d. variable declaration

Q2: True / False

True or False
The new operator creates an instance of a class.

True or False
Each instance of a class has its own set of instance fields.

True or False When you write a constructor for a class, it still has the default constructor that Java automatically provides

Q3. Short Answer

1. What is the difference between a class and an object? Describe the difference in your own words. Give an example.

2. What is the purpose of the “new” keyword?

3. When the same name is used for two of more methods ins the same class, how does Java tell them apart?

Q4: Programming

1. *Square Class*
   Consider the following class declaration:

```
public class Square
{
      private double sideLength;

      public double getArea()
      {
            return (sideLength * sideLength);
      }

      public double getSideLength()
      {
            return sideLength;
      }
}
```

   a. Write a no-arg constructor for this class. It should assign the sideLength field the value 0.0.

   b. Write an overloaded constructor for this class. It should accept an argument that is copied into the sideLength field.

   c. Write a getPerimeter method for this class. It should return the sideLength member multiplied by 4.

-

2. *Circle Class*
   Write a Circle class that has the following fields:
   - radius: a double
   - PI: a final double initialized with the value 3.14159

   The class should have the following methods:
   - Constructor. Accepts the radius of the circle as an argument.
   - Constructor. A no-arg constructor that sets the radius field to 0.0.
   - setRadius. A mutator method for the radius field.
   - getRadius. An accessor method for the radius field.
   - getArea. Returns the area of the circle, which is calculated as area = PI * radius * radius
   - getDiameter. Returns the diameter of the circle, which is calculated as diameter = radius * 2
   - getCircumference. Returns the circumference of the circle, which is calculated as
   - circumference = 2 * PI * radius

   Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

3. *Car class*
   Write a class named Car that has the following fields:
   - yearModel: The yearModel field is an int that holds the car's year model.
   - make: The make field references a String object that holds the make of the car.
   - speed: The speed field is an int that holds the car's current speed.

   In addition, the class should have the following constructor and other methods.
   - Constructor: The constructor should accept the car's yearModel and make as arguments. These values should be assigned to the object's yearModel and make fields. The constructor should also assign 0 to the speed field.
   - Accessors: Appropriate accessor methods (getters and setters) should get the values stored in an object's yearModel, make, and speed fields.
   - accelerate: The accelerate method should add 5 to the speed field each time it is called.
   - brake: The brake method should subtract 5 from the speed field each time it is called.