**Mid-Term Exam #1:**

**Course:** Introduction to Programming  
**Date:** [Thursday (3/7)]  
**Duration:** 60 minutes

#### 

#### ****Part A: Multiple Choice Questions (2 points each)****

**Problem 1:** What does the length() method of the String class return?

1. The memory size of the string
2. The number of characters in the string
3. The first character of the string
4. The last character of the string

**Problem 2:** If String str = "Java Programming";, what will str.substring(5) return?

1. Java
2. Programming
3. Java Prog
4. gramming

**Problem 3:** Which of the following demonstrates a nested if-else statement?

1. if (condition1) { ... } else if (condition2) { ... }
2. if (condition1) { if (condition2) { ... } }
3. if (condition1) { ... } else { ... }
4. if (condition1 && condition2) { ... }

**Problem 4:** Which keyword is used to create a new object in Java?

1. class
2. new
3. struct
4. create

#### Problem 5: What is a constructor in Java?

#### A method that is called to destroy an object

#### A static method that returns a class instance

#### A block of code used to initialize an object

#### An interface that classes must implement

**Problem 6:** If a class Bicycle has a method changeGear, which option correctly calls this method on a Bicycle object named myBike?

1. Bicycle.changeGear();
2. changeGear(myBike);
3. myBike.changeGear();
4. Bicycle.myBike.changeGear();

**Problem 7:** What will be the output of the following Java code snippet?

String test = "Hello, World!";

System.out.println(test.charAt(7));

1. H
2. e
3. W
4. ,

**Problem 8:** What is the output of the following code if x = 5 and y = 3?

if (x > 4) {

if (y > 2) {

System.out.println("X and Y are greater.");

} else {

System.out.println("X is greater.");

}

} else {

System.out.println("X is not greater.");

}

1. X and Y are greater.
2. X is greater.
3. X is not greater.
4. No output.

**Problem 9:** Which principle of object-oriented programming is demonstrated by allowing a class to have more than one constructor method?

1. Inheritance
2. Encapsulation
3. Polymorphism
4. Abstraction

**Problem 10:** When defining a class in Java, what does the private keyword indicate about a member variable?

1. The variable can be accessed from any class.
2. The variable can only be accessed within its own class.
3. The variable can only be accessed within classes in the same package.
4. The variable is a constant and cannot be changed.

#### ****Part B: True/False (2 points each)****

#### ****Problem 1:**** Nested if-else statements can be used to execute different blocks of code based on multiple conditions. **(True/False)**

#### ****Problem 2:**** Constructors in Java can have different names as long as they are within the same class. **(True/False)**

#### ****Problem 3:**** Overloading a method means changing the method's return type. **(True/False)**

#### ****Problem 4:**** The substring(int beginIndex, int endIndex) method includes the character at beginIndex but excludes the character at endIndex. **(True/False)**

#### ****Problem 5:** Java automatically converts the integer to float when an integer variable is assigned to a float variable.**(**True/False)**

#### ****Part C: Short Answer (3 points each)****

#### ****Problem 1:** What is the purpose of the** public static void main(String[] args) **method in Java? The purpose of this method in Java is to serve as the entry point for a program or application.**

#### ****Problem 2:** Is it possible to modify a string using the** toUpperCase() **method? It is possible to modify a string using the method toUpperCase() as it converts all characters from the string to uppercase, so the value of the string is mutated.**

#### ****Part D: Code Implementation (7 points each)****

#### Problem 1: Develop a Java program to implement a class named Circle with a property radius. Include a constructor to initialize the radius, methods to calculate and return the circle's area and circumference. Also, implement a method displayDetails() to print the area and circumference of the circle. In the main method, create a Circle object, invoke its methods, and display its details.

*/\*\*  
 \* Represents a Circle  
 \* @author Canhui Huang  
 \*/*public class Circle {  
 private double radius;  
  
 */\*\*  
 \* Constructs a new Circle  
 \* @param radius the radius of the circle  
 \*/* Circle(double radius) {  
 this.radius = radius;  
 }  
  
 */\*\*  
 \* Computes and returns the area of the circle  
 \* @return the area of the circle as a double  
 \*/* public double area() {  
 return Math.*PI* \* this.radius \* this.radius;  
 }  
  
 */\*\*  
 \* Computes and returns the circumference of the circle  
 \* @return the circumference of the circle as a double  
 \*/* public double circumference() {  
 return Math.*PI* \* this.radius \* 2;  
 }  
  
 */\*\*  
 \* Display details of the circle such as area and circumference  
 \*/* public void displayDetails() {  
 System.*out*.println("Area: " + this.area());  
 System.*out*.println("Circumference: " + this.circumference());  
 }  
  
 public double getRadius() { return this.radius; }  
 public void setRadius(double radius) { this.radius = radius; }  
  
 */\*\*  
 \* Program used to invoke Circle's methods, and display its details  
 \*/* public static void main(String[] args) {  
 Circle circle = new Circle(4.5);  
  
 System.*out*.println("this.area(): " + circle.area());  
 System.*out*.println("this.circumference(): " + circle.circumference());  
 System.*out*.println("\nDisplaying details with this.displayDetails():");  
 circle.displayDetails();  
 }  
}

**Problem 2:** Design a Java program featuring a TrafficLight class to simulate a traffic light at a specific time. The class should include a method showLightColor which takes an integer representing minutes and displays the traffic light color at that time using if-else conditions. Assume the light changes every 5 minutes in the order: Red, Green, Yellow.

To solve the Traffic Light problem, consider these hints:

1. **Understand the Time Cycle:** Recognize that the traffic light changes in a fixed pattern and time frame. Think about how this pattern repeats over time.
2. **Modulo Operator:** Use the modulo operator (%) to determine the current phase within the traffic light cycle based on the given time.
3. **Conditional Logic:** Apply if-else statements to map the calculated phase to the corresponding traffic light color.
4. **Parameter Usage:** The method showLightColor takes time as a parameter. Use this value to decide the traffic light color at the given time.

*/\*\*  
 \* Represents Traffic Light  
 \* @author Canhui Huang  
 \*/*public class TrafficLight {  
 public void showLightColor(int minute) {  
 // Minute [0,5): Red  
 // Minute (5,10): Green  
 // Minute (10,15): Yellow  
 // The cycle pattern repeats every 15 minutes  
  
 System.*out*.print("At minute " + minute + ". ");  
 if ( minute % 15 >= 0 && minute % 15 < 5) {  
 System.*out*.println("Traffic light color: Red");  
 } else if ( minute % 15 < 10 ) {  
 System.*out*.println("Traffic light color: Green");  
 } else {  
 System.*out*.println("Traffic light color: Yellow");  
 }  
 }  
  
 */\*\*  
 \* Java program featuring a TrafficLight class to simulate a traffic light at a specific time  
 \*/* public static void main(String[] args) {  
 TrafficLight trafficLight = new TrafficLight();  
  
 trafficLight.showLightColor(2);  
 trafficLight.showLightColor(6);  
 trafficLight.showLightColor(12); System.*out*.println();  
  
 trafficLight.showLightColor(17);  
 trafficLight.showLightColor(21);  
 trafficLight.showLightColor(28); System.*out*.println();  
  
 trafficLight.showLightColor(0);  
 trafficLight.showLightColor(5);  
 trafficLight.showLightColor(10);  
 trafficLight.showLightColor(15);  
 }  
}