Arab Academy of Science and Technology and Maritime Transport

COLLEGE OF ENGINEERING & TECHNOLOGY

Computer Engineering Department

Course: **Object Oriented Programming**

Course Code: **CC316**

Sheet No.: **4**

1. Knowing that class Circle extends GeometricObject, answer the following questions:
   a. Assume are circle and object1 created as follows:

      Circle circle = new Circle(1);
      GeometricObject object1 = new GeometricObject();

      Are the following Boolean expressions true or false?

      (circle instanceof GeometricObject)
      (object instanceof GeometricObject)
      (circle instanceof Circle)
      (object instanceof Circle)

   b. Can the following statements be compiled?

      Circle circle = new Circle(5);
      GeometricObject object = circle;

   c. Can the following statements be compiled?

      GeometricObject object = new GeometricObject();
      Circle circle = (Circle)object;

2. Write the following method that returns the maximum value in an **ArrayList** of integers. The method returns **null** if the list is **null** or the list size is **0**.

   public static Integer max(ArrayList<Integer> list)

3. Write the following method that sorts an **ArrayList** of numbers:

   public static void sort(ArrayList<Integer> list)

4. Design a new Triangle class that extends the abstract GeometricObject class. Draw the UML diagram for the classes Triangle and GeometricObject and then implement the Triangle class. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not

5. **Design and implement the classes that represent a simple employee system at a company:-**

class Employee

- Each employee in the company has variables representing his name and salary.
- Each employee has accessor method for name & salary and a mutator method for salary
- Each employee has method calculateSalary() that calculates & sets the value of salary.
- An employee can be either a full time employee representd by class FullTimeEmployee or a part time employee represented by class PartTimeEmployee
- No objects should be created directly from class Employee

class FullTimeEmployee

- *Each full time employee has a basic salary variable initialized to 1000 (basicSalary) and a variable representing his employment years (employmentYears)*
- Each object of class FullTimeEmployee is initialized using the name of the employee .
- Class FullTimeEmployee has the following methods :

- Accessor and mutator methods for employmentYears
- *method calculateSalary() set the salary equal to the basicSalary+ employmentYears*100*

class PartTimeEmployee

- *Each part time employee has a variable representing hours worked (hours) and another variable representing his hourly wage (wage).*
- *Each object of class PartTimeEmployee is initialized using all of its instance variables.*
- Class PartTimeEmployee has the following methods :
- *Accessor and mutator methods for instance variables*
- method calculateSalary() set the salary equal to the hours*wage.

Test your code and Draw the UML class diagram

6. Write a class named **GroceryList** that represents a person's list of items to buy from the market, and another class named **GroceryItemOrder** that represents a request to purchase a particular item in a given quantity (example: four boxes of cookies).

The **GroceryList** class **has a** list of grocery item orders and keeps track of its size (number of items in the list so far). Assume that a grocery list will have no more than 10 items. A **GroceryList** object should have the following methods and constructor:.
public GroceryList()

constructs a new empty grocery list.
public void add(GroceryItemOrder item)

adds the given item order to this list, if the list is not full (i.e., has fewer than 10 items).
public double getTotalCost()

returns the total sum cost of all grocery item orders in this list.
The **GroceryItemOrder** class should store an item name, quantity and a price per unit. A **GroceryItemOrder** object should have the following :
public **GroceryItemOrder** (String name, double pricePerUnit)

constructs an item order to purchase the item with the given name, in the given quantity, which costs the given price per unit.
public double getCost()

returns the total cost of this item in its given quantity. For example, four boxes of cookies that cost 2.30 per unit have a total cost of 9.20.
public void setQuantity(int quantity)

sets this grocery item's quantity to be the given value.
Test your code.

7. Design an interface named Colorable with a void method named howToColor() . Every class of a colorable object must implement the Colorable interface. Design a class named Square that extends GeometricObject and implements Colorable. Implement howToColor to display the message Color all four sides.

   Draw a UML diagram that involves Colorable, Square, and GeometricObject. Write a test program that creates an array of five GeometricObjects. For each object in the array, display its area and invoke its howToColor method if it is colorable.