

**CS217 Object Oriented Programming**  
**Assignment # 5 (Part 1)**  
**(Composition, Aggregation, Inheritance, and Polymorphism)**

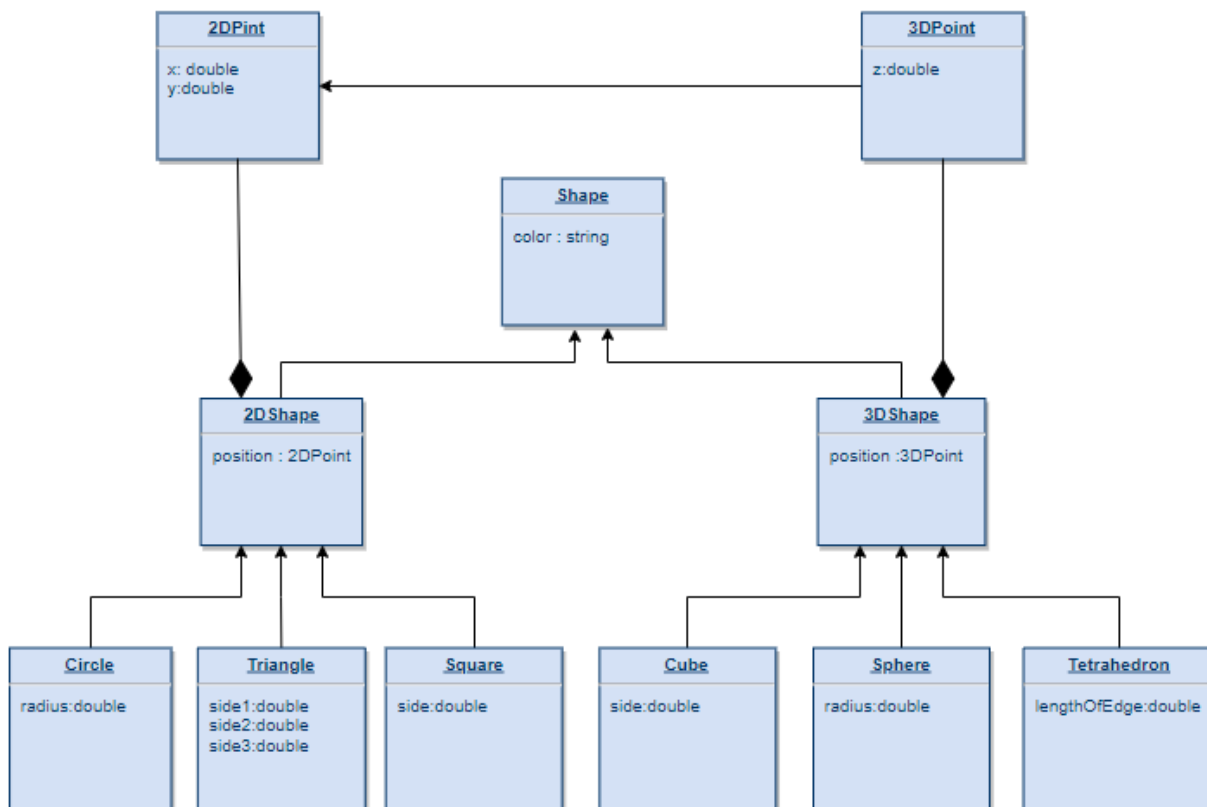
**Deadline: 22 April 2019**

**Attention**

- Make sure that you read and understand each and every instruction. If you have any questions or comments, you are encouraged to discuss with your instructors (and colleagues) on Piazza.
  - Plagiarism is strongly forbidden and will be very strongly punished. If we find that anyone has copied from someone else, with or without his or her knowledge, we may award the involved parties a straight zero in the assignment (or all the assignments).
  - **Note: We will be running your code against our test cases, and a test case failure or a segmentation fault/incorrect result will result in loss of marks.**
- 

**Question 1:**

In this question, you are required to develop a host of classes, and place them in a reasonable hierarchy, as shown below:



## Description

For the *Shape* class hierarchy, create a *Shape* class that will be the base class of all the other shapes. It should have one private string instance variable representing the shape's color, and should have the following public member functions:

```
Shape() //Default constructor
Shape(const string& color) // a constructor that sets the color instance value.
string getColor() const //a const member function returning the object's color
void setColor(const string& color)//a member function setting the object's color
value.
string toString() //function that returns the shape's description (color,
type,position, measurements, and area) as a string. It must be overridden in each
derived class.
```

Create a *2DPoint* class. It should have two private double instance variable representing the x and y coordinates of position and should have the following public member functions:

```
2DPoint() //Default constructor
2DPoint(double, double) //Parameterized constructor
double getX() const //a const member function returning the x coordinate value
void setX(double) //a member function setting the x coordinate value
double getY()const // a const member function returning the y coordinate value.
void setY(double) //a member function setting the y coordinate value.
```

Create a *3DPoint* that is derived from *2DPoint*. It should have a private double instance variable representing the z coordinate of position and should have the following public member functions:

```
3DPoint() //Default constructor
3DPoint(double,double, double) //Parameterized constructor that invokes the base
2DPoint class to set x,y coordinates values then sets its z dimension.
double getZ()const // a const member function returning the Z coordinate value.
void setZ(double) //a member function setting the Z coordinate value.
```

Create a *2DShape* class that is derived from *Shape*. It should have one private *2DPoint* instance variable representing the position of 2D shape, and should have the following public member functions:

```
2DShape() //Default constructor
2DShape(const string& color , double x, double y) //Parameterized constructor that
invokes the base Shape constructor to set color then sets its own coordinates
instance value.
double area() //member function that computes and returns the object's area. It must
be overridden in each derived class.
```

```
double perimeter(); //member function that computes and returns the object's
perimeter. It must be overridden in each derived class.
```

Create a 3DShape class that is derived from Shape. It should have one private 3DPoint instance variable representing the position of 3D shape, with a default and parametrized constructor and member functions for volume.

Create a Circle class that is derived from 2DShape. It should have one private double instance variable representing the radius, and should have the following public member functions:

```
Circle(const string& color, double x, double y , double radius) //a constructor
that invokes the base 2DShape constructor then sets its own radius instance value.
double area() //this overriding member function computes and returns the
Circle object's area value.
double perimeter () //this overriding member function computes and returns the
Circle object's perimeter value.
string toString() //this overriding member function returns the Circle
object's description ( color, type, measurements, perimeter and area) like
Red Circle Position:(x,y) Radius:Value1 Perimeter:Value2 Area:Value3
```

Create a Square class with one private double instance variable representing the side length, with a default and parametrized constructor and overriding member functions for area(), perimeter() and toString().

Create a Traingle class with two private double instance variable representing the width, length, with a default and parametrized constructor and overriding member functions for area(), perimeter() and toString().

Create a Sphere class that is derived from 3DShape. It should have one private double instance variable representing the radius, and should have the following public member functions:

```
Sphere(const string& color, double x, double y, double z, double radius)//a
constructor that invokes the base 3DShape constructor then sets its own radius
instance value.
double voulme() //this overriding member function computes and returns the
Spehere object's volume value.
string toString()//this overriding member function returned the object's
description ( color, type, measurements, volume) like
Red Sphere Position:(x,y,z) Radius:Value1 Vloume:Value3
```

Create a Cube class with one private double instance variable representing the side length, with a constructor and overriding member functions for volume() and toString().

Create a Tetrahedron class with one private double instance variable representing the length of edge, with a constructor and overriding member functions for volume() and toString().

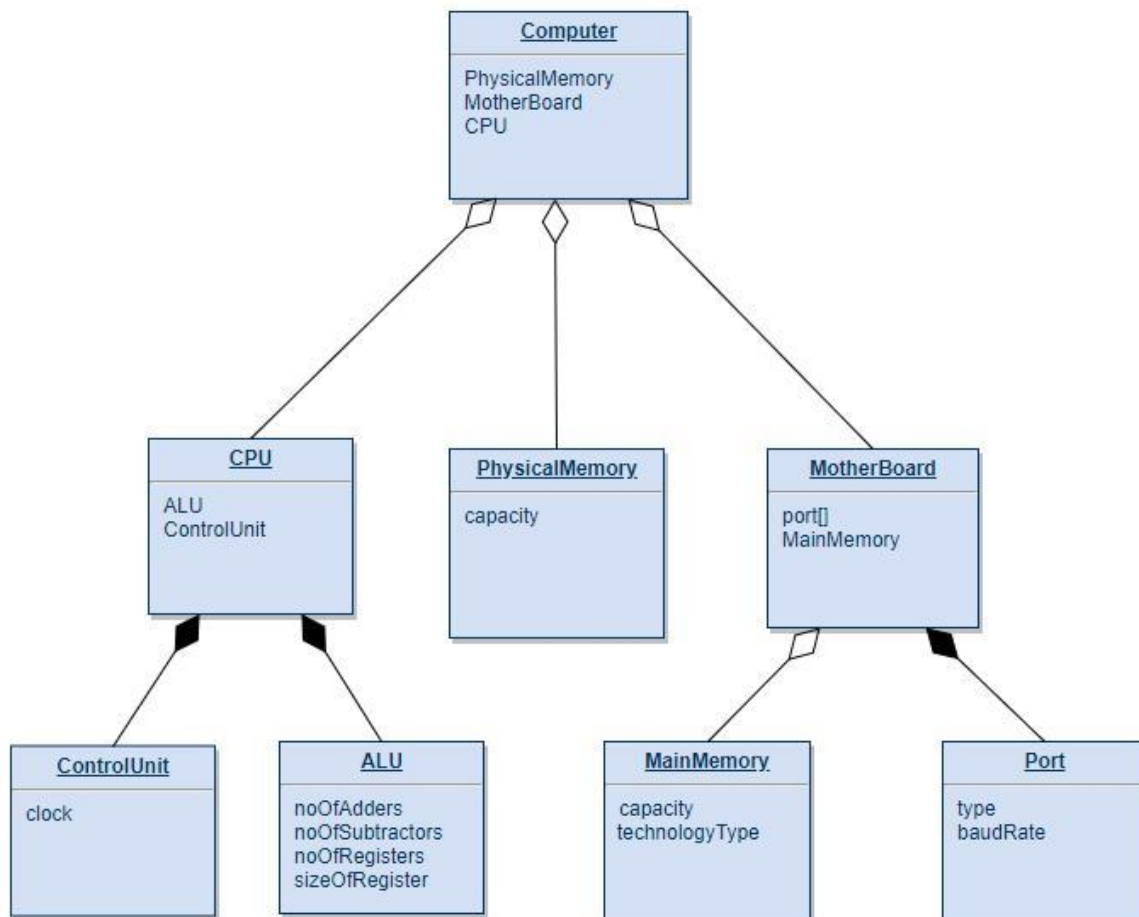
Each derived class constructor must use the constructor initializer syntax to call the base Shape constructor.

To check shape data, create a ShapeFactory.cpp file containing a getShape() function and polymorphic arrays of type 2DShape and 3DShape. It should read a shape description from the input stream, create the correct type of derived shape with the new operator and parameters to the constructor. After reading a shape type (e.g. circle), it reads the additional information specific to that type of shape (e.g. for a circle, it reads the radius position and color), and then uses the new operator to create the specific derived type of shape (e.g. new Circle(radius) ) and add to the dynamic array of the specific type.

Initially the size of each Array will be 1 and it will point to NULL. After creating a new circle add it to 2DShape Array by increasing the capacity of the circle Array by 1. The last index will always point to Null indicating to the end of the Array.

## Question 2:

In this question, you need to write a host of classes, and place them in a reasonable hierarchy as shown below:



**Description:**

Design a class ALU which include the following attributes:

- NoOfAdders: a int
- NoOfSubtractor: a int
- NoOfRegisters: a int
- sizeOfRegisters: a int

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data members.

Design a class ControlUnit which includes the following:

- clock: a float

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data member as given below.

Design a class CPU which is composed of ALU and CU. Data member are:

- alu: a ALU
- cu: a ControlUnit

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data members.

Design a class MainMemory which includes the following:

- capacity: an int
- technologyType: a string (Possible values :Semiconductor, Silicon)

□

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data member.

Design a class Port which includes the following:

- type: a string (Possible values : VGI Port,I/O Port,USB Port,HDMI Port etc)

- baud\_rate: an int

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data member.

Design a class MotherBoard which is composed of Ports (IO ports, VGI ports etc) and aggregated with

MainMemory:

- mm: A MainMemory
- ports: ports array

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data members.

Design a class PhysicalMemeory which includes the following:

- capacity: an int

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data member.

Design a class Computer which is aggregated of PhysicalMemory, CPU and MotherBoard includes the following:

- pm: A PhysicalMemory
- mb: A MotherBoard
- cpu: A CPU

The class has the following member functions.

1. A constructor initializing the number with default parameters.
2. A constructor initializing the number with Overloaded Constructors.
3. Getters and Setters of the class data member.

To check the scenario, create a Shop class which manufacture and sell the Computers. In Shop class write a function manufactureComputer() which create a new Computer Object by taking all necessary specifications from user and add it to a Dynamic Array of Computers. After Successful Computer Object Creation add it to the List by adjusting the capacity of Array. The User can view List of Computers created.

### Question 3:

In this Question you will design an automated management system for a Walk-through Restaurant where customers can place their orders and pay the bill to receive ordered items. Read the case study carefully and identify all the classes necessary to develop the system and put them in a reasonable hierarchy.

The system enable the user to view the orders placed by Customers. For each order the system stores customer information, Date and Time at which the order was placed, Total bill and status (pending, in-progress, Ready).For each Order 5% Tax is included in the grand total.

Customers can put orders by selecting items from the list of available items, for this, the restaurant must use customer's name and phone number. User can Place Order and make Payment in the system. The restaurant menu is organized in categories (Main course, Beverages, and Desserts) of menu items. Each menu item has an id, name, and price in Rs.

The Main Course Menu includes Sandwiches, Pizzas and a further sub-category of Fried Items. For each main course item customer can provide a description (Special Instructions) .Sandwich can be specified by its filling (Smoked Chicken, BBQ Chicken, Chicken Jalapeno, Vegetable) and type (Grilled/Plain).The restaurant offers Pizza in four different sizes (small, medium, large, extra-large) with three different crust options (Fresh pan, Stuffed and Hand-tossed) and many topping options (Pepperoni, Mushrooms, Onions, Extra cheese and Black olives).

The Fried items sub-category include Nuggets, Fries and Burgers. Each Fried Item is available with many sauce options to choose from. The restaurant offers three different types of nuggets (Chicken Nuggets, Tempura Nuggets and Crispy Chicken Nuggets) in three different shapes (circle, star and heart) with dipping Sauces (Sweet and Sour, Honey Mustard, Sweet Chili and Hot Mustard) .Two types of fries are available: curly and plain in three different servings (Small, Regular, Jumbo) with different sauces (Garlic-Mayo, Chocolate and Cheddar Cheese). To order a Burger a customer can select from different options available for Patties (Beef, Crispy Chicken, Grilled Chicken and fish) and Bun Type (plain and sesame seed).The Burger Sauces (Jalapeno cheese sauce, Classic burger sauce, cheese mayo and Truffle mayo).

The restaurant offers Beverages of different flavors in three different servings (Small, Regular, Large) and in two different temperature levels ( i.e. Room temperature, Chill).Beverages category is further divided in two sub categories (Carbonated Drinks, Fresh Juices).

Carbonated Drinks (Sprite, 7up, Pepsi, Coke, Red Bull and Fanta) can be specified by name and whether or not the drink is sugar free. Fresh Juices have two types (Fresh/Pre Packed) and can further be specified whether or not it's seasonal.

Dessert menu contains a variety of flavors of Ice Creams and Cookies. The System stores Price per scoop for all available ice creams. Ice Cream Price is calculated with Numbers of scoops and Customer can choose from two types to toppings (Wafers and Peanuts).Cookies are available in Different Flavors (Chocolate chip, Peanut, Coconut). The System Store price per gram for each Cookie type. The price of the Cookies is calculated by price per gram and is also available as Sugar free. Cookies are Available in Different shapes (Round, Square).

The System take user order by Displaying the Menu to User. Each order can be composed of multiple items and the system generate and display the total bill to the user. User can then proceed with the order or cancel the order. If user choose to proceed with the order the system display the Bill. User can then pick the order from the restaurant by paying bill. System should be able to display all the cancelled Orders and Ready Orders.

**Note:** Create a polymorphic array of Items inside the Order Class and dynamically increase the capacity of the array as you have done in first question inside shape Factory Class.

Payment is handled manually outside the system. System will only Display the grand Total to the Customer. When the User Successfully complete an order the system change the status to ready immediately and Displays a Success Message to customer.