

# Assignment – Pillars of OOP

Weightage: 10 marks

## Note:

- Plagiarized tasks will be awarded zero marks
- Submit cpp files with proper naming.

## 1. The Art of Zodiac Signs through OOP Composition



Are you curious about your Zodiac sign and want to learn more about the mysteries of the stars? Do you have a passion for coding and want to use your skills to create a personalized Zodiac tool? If so, this assignment is for you!

In this assignment, you will be implementing composition in object-oriented programming using the example of a **Person**, their **DateOfBirth**, and their **Zodiac**. You will design classes for each of these objects, with each class containing interesting methods and variables. You must create the class diagram for this task.

Class: **Zodiac**

Variables:

name (string): representing the name of the person's zodiac sign (eg: Aries, Leo, etc)

description (string): a description of the person's zodiac sign (personality traits).

Methods:

void generateZodiacData(day, month): that takes the day and month of a person and find the zodiac sign against it, by assigning the name, and the relevant description to the variables.

void displayPerosnality(): By calling this method, display the name of the zodiac, along with its personality, by simply printing the instance variables of that class.

Class: **DateOfBirth**

Variables:

day (integer): representing the day of the month of the person's birth

month (integer): representing the month of the person's birth

year (integer): representing the year of the person's birth

Object of the class Zodiac

Class: **Person**

Variables:

name (string): representing the person's name

date\_of\_birth (DateOfBirth): an instance of the DateOfBirth class

Methods:

displayZodiacInfo()

**Note:** Make getter, setter methods, constructors, where required. Now think how to ultimately call the displayPerosnality() method indirectly with the object of person.

In main there should be only object of person, and initialize it. Do not violate the principles of composition.

### Sample Output:

```
Enter your name:
Fariba
Enter your date of birth:
Day: 28
Month: 2
Year: 1999

Fariba, Your Zodiac sign with persoanlity traits are:

Zodiac sign: Pisces

Personality: Compassionate, imaginative, emotional, artistic, and dreamy
They have a creative and intuitive nature, with a deep sense of compassion and empathy for others.
They love art and music, and live in their own fantasy world.
-----
```

---

## 2. From Eye in the Sky to Cargo Transport: Creating a Versatile UAV"



You are working for a UAV manufacturer company that has designed a basic UAV model with some common features.

### UAV Class

Methods:

takeOff(), land()

The company now wants to create two variants of the basic model to target different markets - one for surveillance and the other for cargo delivery. Both variants has now their own additional properties and behaviours.

### **SurveillanceUAV Class**

Attributes:

Name, weight, maxAlitude, zoomLevel, recordingCapacity, isNightVision

Methods:

captureImage(), recordVideo()

### **CargoDeliveryUAV**

Attributes:

Name, weight, maxAlitude, cargoArea

Method:

deliverCargo()

The company has decided to create a third UAV model that combines the features of both variants and can perform both surveillance and cargo delivery missions. This model is called the "Multi-Purpose UAV" and it should inherit from both the surveillance and cargo delivery variants. This variant has a solar panel attached to it for high power, as it has to deliver and also do the surveillance task.

### **Class MultiPurposeUAV**

Attributes:

Name, weight, maxAlitude, solarPanelEfficiency;

Methods:

deliverCargoWithSurveillance()

This method will call the methods in the following order.

takeOff()

recordVideo()

land()

deliverCargo()

In main, create objects of SurveillanceUAV, CargoDeliveryUAV, and MultiPurposeUAV. Call the display() method of the objects. Plus call the deliverCargo() with CargoUAV object, recordVideo() with SurveillanceUAV object, and deliverCargoWithSurveillance() with multi functional UAV object.

**Note:** Resolve the diamond problem for land() and takeOff() methods in this task. Create a class diagram too for that task.

**Sample Output:**

```
I am a Cargo Delivery UAV
My name isMulti purpose drone
My weight is: 14.5 kg
My altitude is: 3000.000 ft
My cargo area is: 40.000
Delivering cargo

I am a Survellience UAV
My name is: Survellience drone
My weight is: 16.700 kg
My altitude is: 1000.00 ft
My zoom level is: 20.00 x
My recording capacity is: 3.00 GB
I have a night vision
Recording video

I am a Multi Purpose UAV
My name is: Multi purpose drone
My weight is: 10.400 kg
My altitude is: 2000.000 ft
My zoom level is: 20.000 x
My recording capacity is: 5.000 GB
I have a night vision
My cargo area is: 5.000
My solar panel efficiency is: 5.000

I am going to deliver a cargo along with survellience
Taking off
Recording video
Landing
Delivering cargo
```

---

### 3. "The Polymorphic Arena: A Battle of Character Types"

"Are you ready to embark on a thrilling Role Playing Game adventure? In this task, you'll enter a world of fantasy and imagination as you create a set of characters for a battle royale like no other. The game consists of warriors, mages, and thieves, each with their own unique abilities and skills. Your task is to create a set of classes using polymorphism that models the different types of characters and their attacks



Warriors are strong and attack with swords, mages are intelligent and attack with fireballs, and thieves are agile and attack with daggers.

The Character class is the superclass of the three character types and has a name instance variable and an empty attack() method. The subclasses Warrior, Mage, and Thief inherit from the Character class and each one implements its own unique version of the attack() method, as well as an additional instance variable (strength for Warrior, intelligence for Mage, and agility for Thief).

The attack method simply prints the message for each character type e.g.: Attacks with intelligence for Mage.

Your task is to create a Main method that creates instances of each character type and calls their attack() method using polymorphism. Each character's unique ability should also be displayed along with their name. For example, "John attacks with strength 10" for a warrior named John.

### Sample Output:

```
Enter the Warrior name:
Jhon
Enter the Warrior strength:
10

Enter the Mage name:
Edward
Enter the Mage intelligence:
20

Enter the Thief name:
Chris
Enter the thief agility:
15

Jhon attacks with strength 10
Edward attacks with fireballs using intelligence 20
Chris attacks with daggers using agility 15
-----
```

---