# Lab Assignment # **10**

---

**Lab Objectives.** The goal of this lab is to help you practice **Inheritance** and **Polymorphism.**

**Submitting Your Solutions.**

- All solutions must be submitted through the e-Learning system.
- The lab duration is 3 hours. You can receive support from your TA during this time.
- Submissions after 4 hours will note be accepted.

---

📷 *You are required to open the camera, **share your screen** with a TA and show her your progress **at least once** during the lab session.*

✋ ***Asking questions** during the lab is allowed and is encouraged.*

📖 *This is an **open-book** assignment. You are strongly encouraged to consult your notes when working on the exercises.*

🚫 *You are not allowed to **look** at the code of others or **dictate** any code to anyone during the lab. You are not allowed to **show** anyone (except your instructor) any part of your code during the lab.*

---

❗ **Note.** There could be a deduction of a maximum of 5 points for not adhering to the course style guidelines (they are provided at the end of the handout for reference).

# Before Starting

Download and study :

1. **Run** the given program to understand its behavior.

2. **Read** through the program and think carefully about the following questions:
   a. Which parts of the 3 classes are common to all of them?
   b. How can we fix the code duplication in the 3 classes?
   c. Which parts of `main()` repeat?
   d. Can the program use the three classes and have the same behavior but without code duplication?

# Refactoring The Classes (66 points)

Fix the code duplication in the three Question classes using **inheritance.** Your solution is expected to contain a single base class named **Question** that is *abstract* (with at least *one pure virtual* function).

You will be graded based on the following:

- The choice of the **data members** in the base class.

- The choice and implementation of the **constructors** in the base class and derived classes.

- The choice and implementation of the **non-pure functions** in the base class.

- The choice of the **pure virtual function(s)** in the base class and its/their implementation in the derived classes.

- How clean your design is from an OOP perspective.

# A Class for Math Questions (34 points)

Fix the code duplication in `main()`. Your solution is expected to create an **array of 6 Question** objects and loop through the array and use *polymorphism* to perform exactly the same thing the original code performs.