# Object Oriented Programming – Fall 22
# (BS-CS-F22)
# Lab-9
# Lab Instructor: Maa'm Sanam Ahmad

**Instructions:**

- **Indent your code properly.**
- **Use meaningful variable and function names.**
- **Use the camelCase notation.**
- **Use meaningful prompt lines/labels for all input/output.**
- **Do NOT use any GLOBAL variable(s). However, global named constants may be used.**
- **This is an individual lab, you are strictly NOT allowed to discuss your solution with fellow colleagues, even not allowed to ask how is he/she is doing, it may result in negative marking. You can ONLY discuss with your TAs or with me. • Anyone caught in an act of plagiarism wouldbe awarded an "F" grade in this Lab.**

Do Validations on inputs where required otherwise 1 mark will be deducted for every wrong validation.

## Tasks:

*1.1* Implement a class named **Employee**. The class should hold the following information in its private member variables:

- o Employee name (a c-string which can contain at most 50 characters)
- o Employee number (an integer)

You are required to implement the following public member functions of the **Employee** class:

- A parameterized (overloaded) constructor (which takes 2 **Parameters**)
- Appropriate getter and setter functions
- A function named **display** to display the details of an Employee on screen
- Copy constructor
- Overloaded assignment operator

*1.2* Next, implement a class named **ProductionWorker** that is derived from the **Employee** class. The **ProductionWorker** class should have private member variables to hold the following information:

- o Shift (an integer)
- o Hourly pay rate (a double)

*<u>Note:</u> The workday is divided into two shifts: **day and night**. The shift variable will hold an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2.*

Implement the following member functions for the **ProductionWorker** class:

- A parameterized (overloaded) constructor (which should take **4 Parameters**)
- Appropriate getter and setter functions

- A function named **display** to display all details of a **ProductionWorker** on screen
- Copy constructor
- Overloaded assignment operator

***1.3*** Demonstrate the above classes and their member functions by writing a program that creates two **Employee** objects (one on the stack and one on the heap) and two **ProductionWorker** objects (one on the stack and one on the heap). All the details of these objects must be taken from the user.

***1.4*** In a particular factory, a team leader is a production worker who leads a small team of workers. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Implement a **TeamLeader** class that inherits from the **ProductionWorker** class which you implemented above in **Task # 1.2**. The **TeamLeader** class should have private member variables to hold the following information:

- o Monthly bonus amount (a double)
- o Required number of training hours (an integer)
- o Number of training hours that the team leader has attended (an integer)

Implement the following member functions for the **TeamLeader** class:

- A parameterized (overloaded) constructor (which should take 7 Parameters)
- Appropriate getter and setter functions
- A function named **display** to display all details of a **TeamLeader** on screen
- Copy constructor
- Overloaded assignment operator

***1.5*** Enhance your driver program by implementing the following functionalities:

Ask the user that how many **TeamLeader** objects he/she wants to create (let's say **n**).

Dynamically allocate an array of **TeamLeader** ***pointers*** of size **n**.

Using a loop, ask the user about all details of each **TeamLeader** object, and allocate that object (on the heap) through the appropriate pointer.

Now, use a loop to display all details of all **TeamLeaders** on screen.

Finally, your program should properly deallocate all dynamically allocated memory.