

# National University of Computer and Emerging Sciences



## Lab Manual 10 Object Oriented Programming

Course Instructor	Ms. Arooj Khalil
Lab Instructor (s)	Mamoona Akbar Saleha Batool
Section	BSE-2B
Semester	Spring 2023

Department of Computer Science  
FAST-NU, Lahore, Pakistan

## Objectives:

After performing this lab, students shall be able to understand:

- ✓ Polymorphism
- ✓ Virtual and Pure virtual functions
- ✓ Abstract classes

## TASK 1

Write a program to calculate the area of following shapes by using Public -- Single Inheritance and polymorphism.

The base class is “shape” and the derived classes are rectangle, triangle and circle  
Attributes of all the classes are as under:

shape	rectangle	triangle	circle
<b>protected:</b> string type; <b>public:</b> virtual void area_calculator();	<b>public:</b> void area_calculator() { //definition } <b>private:</b> float height; float width;	<b>public:</b> void area_calculator() { //definition } <b>private:</b> float base; float height;	<b>public:</b> void area_calculator() { //definition }float radius;
	Area = Length X Width	Area = 1/2 of the base X the height	$A = \pi r^2$

1. Your each class must have overloaded/default constructor to initialize required parameters for calculating area. You might have to use dummy arguments in your base class overloaded constructors.
2. Create a function in base class to print the variable “type”.
  - a. Override this function in child class so that it can be called from child classes as well.
3. In main,
  - a. Create objects of derived classes rectangle, triangle and circle.
  - b. Create a pointer of base class “area”.
  - c. With this pointer, point to the objects of derived classes one by one and calculate area of each individual shape.

## TASK 2

### 1.1.

- Create an **abstract** class called Account.
- It has data member:
  - Private Account Number.
  - Private Account Balance.
- Add suitable setter/getter for data.
- Add Debit(float), Credit(float) as member functions (**Pure Virtual**).
- Add Print() function (**Virtual**)
  - Override Print Debit and Credit functions according to derived classes.

### 1.2.

- Create a class called CurrentAccount i-e: CurrentAccount(is-a) Account
- It has data member:
  - Service Charges (To be charged during credit if account balance is less than min balance )
  - Minimum Balance
- Override print() as created in above class which displays:
  - Account Number, Account Balance, Minimum Balance, Service Charges
- Modify the definition of the print() so that it displays a suitable message containing above info.
- Similarly override credit(float), debit(float) functions such that credit(float) simply add amount to the Account Balance and debit(float) checks if the amount to be debited is within the range of Account Balance, and further if the amount is account balance is less than min balance standard charges would also be deducted.
- Create a class called SavingAccount i-e: SavingAccount (is-a) Account
- It has data member:
  - Interest Rate.
- Override print() as created in parent class which displays:
  - Account Number, Account Balance, Interest Rate
- Modify the definition of the print() so that it displays a suitable message containing above info.
- Similarly override credit(float), debit(float) functions such that credit(float) simply add amount to the Account Balance and debit(float) checks if the amount to be debited is within the range of Account Balance.

### 1.3.

- In main function:
  - Create an array of Account type Pointers, of size 3. Assign Account object to index 0, CurrentAccount object to index 1 and SavingAccount object to index 2.
  - Have you encountered any problem? Report the problem and change the size and elements of the array accordingly

- Now Credit and Debit the CurrentAccount and SavingAccount in main function

#### **1.4.**

Although things seem to be fine on the surface, there is a problem in the program we just wrote. To observe this problem, we must add destructors for all classes. Paste the following inline definitions of the destructors in their corresponding classes, execute the program and paste the output below.

Your destructors should delete the child classes first before destroying parent class. There should not be any memory leaks.