National University of Computer & Emerging Sciences, Karachi
Computer Science Department
Fall 2022, Lab Manual - 03

Course Code: CL-1004	Course : Object Oriented Programming Lab
Instructor(s) :	Shahroz Bakht, Ayesha Ali, Zumar Noor

Contents:

Introduction to Classes & Objects

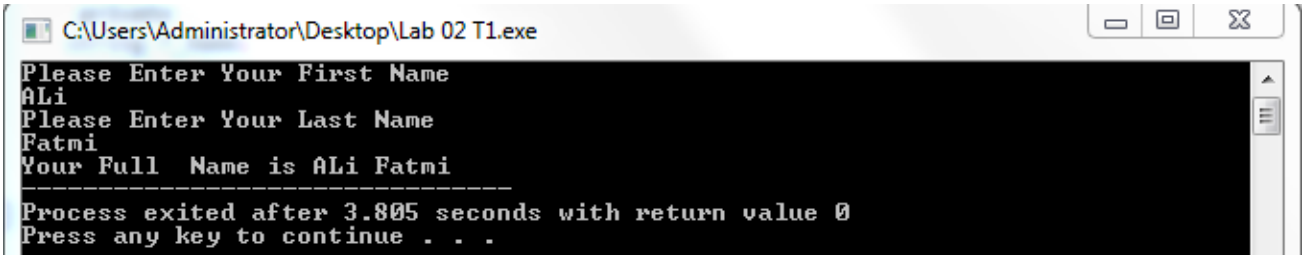
- A ***class*** *is a definition of objects of the same kind*. In other words, a *class* is a blueprint, template, or prototype that defines and describes the *static attributes* and *dynamic behaviors* common to all objects of the same kind.
- A **class** can be visualized as a three-compartment box, as illustrated:
  - ***Classname*** (or identifier): identifies the class.
  - ***Data Members*** or ***Variables*** (or *attributes*, *states*, *fields*): contains the *static attributes* of the class.
  - ***Member Functions*** (or *methods*, *behaviors*, *operations*): contains the *dynamic operations* of the class.
- An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. When a class is defined, only the specification for the object is defined; no memory or storage is allocated. To use the data and access functions defined in the class, you need to create objects.

Sample C++ Code:

Code#1

```
#include<iostream>
using namespace std;

class Studnet
{
   private :
  string F_Name;
   string L_Name;

  public:

    void input_value()
    {
      cout << "Please Enter Your First Name\n";
```

```
        cin >> F_Name;
        cout << "Please Enter Your Last Name \n";
        cin >> L_Name;
      }

      void output_value()
      {
        cout << "Your Full  Name is "<<F_Name<<" " <<L_Name;

      }
};

main()
{
  Studnet object;

  object.input_value();
  object.output_value();

  //object.variable;  Will produce an error because variable is private

  return 0;
}
```



Code#2

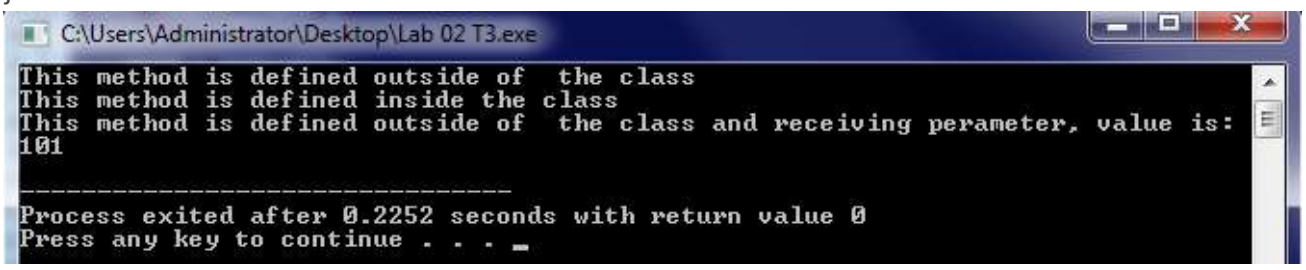
```
#include <iostream>
using namespace std;

class Box {

  public:
   double Lenght;
   double Breadth;
   double Height;

   double Area() {
```

```
        return Lenght * Breadth;
    }

    double Volume() {
        return Lenght * Breadth * Height;
    }
};

int main() {


    Box obj;

    obj.Lenght = 30;
    obj.Breadth = 40;
    obj.Height = 60;


    cout << "Area of Box =  " << obj.Area() << endl;
    cout << "Volume of Box =  " << obj.Area() << endl;

    return 0;
}
```

Code#3

```
#include <iostream>
using namespace std;

class Sample_Class {
public:
   void o_method();
   void o_method2(int value);

   void i_method()
{
   cout << "This method is defined inside the class\n";
}
```

```
};


void Sample_Class::o_method()
{
   cout << "This method is defined outside of  the class\n";
}




void Sample_Class::o_method2(int number)
{
   cout << "This method is defined outside of  the class and receiving perameter, value is: " << number << "\n";
}

int main()
{

   Sample_Class obj;
   obj.o_method();
   obj.i_method();
   obj.o_method2(101);

   return 0;
}
```

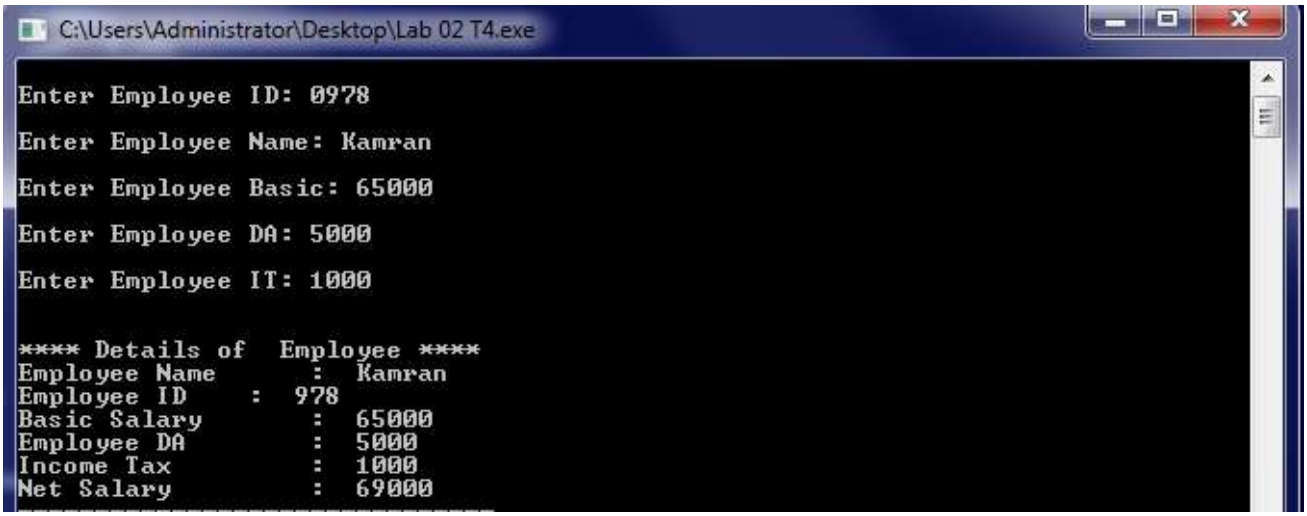


Code#4

```
#include <iostream>
using namespace std;
```

```
class employee
{
        public:
        int   E_id;
        string  E_name;
        float E_basic;
        float E_da;
        float E_it;
        float E_net_sel;

        public:


                float find_net_salary(float basic, float da, float it);
                void show_emp_details();
};


float employee :: find_net_salary(float basic, float da, float it)
{
   return (basic+da)-it;
}

void employee :: show_emp_details()
{
        cout<<"\n\n**** Details of  Employee ****";
        cout<<"\nEmployee Name      :  "<<E_name;
        cout<<"\nEmployee ID    :  "<<E_id;
        cout<<"\nBasic Salary       :  "<<E_basic;
        cout<<"\nEmployee DA        :  "<<E_da;
        cout<<"\nIncome Tax         :  "<<E_it;
        float net_salary=find_net_salary(E_basic, E_da, E_it);
        cout<<"\nNet Salary         :  "<<net_salary;
        cout<<"\n------------------------------\n\n";
}


int main()
{
   employee emp;
   cout<<"\nEnter Employee ID: ";
```

```
        cin>>emp.E_id;
        cout<<"\nEnter Employee Name: ";
        cin>>emp.E_name;
        cout<<"\nEnter Employee Basic: ";
        cin>>emp.E_basic;
        cout<<"\nEnter Employee DA: ";
        cin>>emp.E_da;
        cout<<"\nEnter Employee IT: ";
        cin>>emp.E_it;

    emp.show_emp_details();


    return 0;
}
```



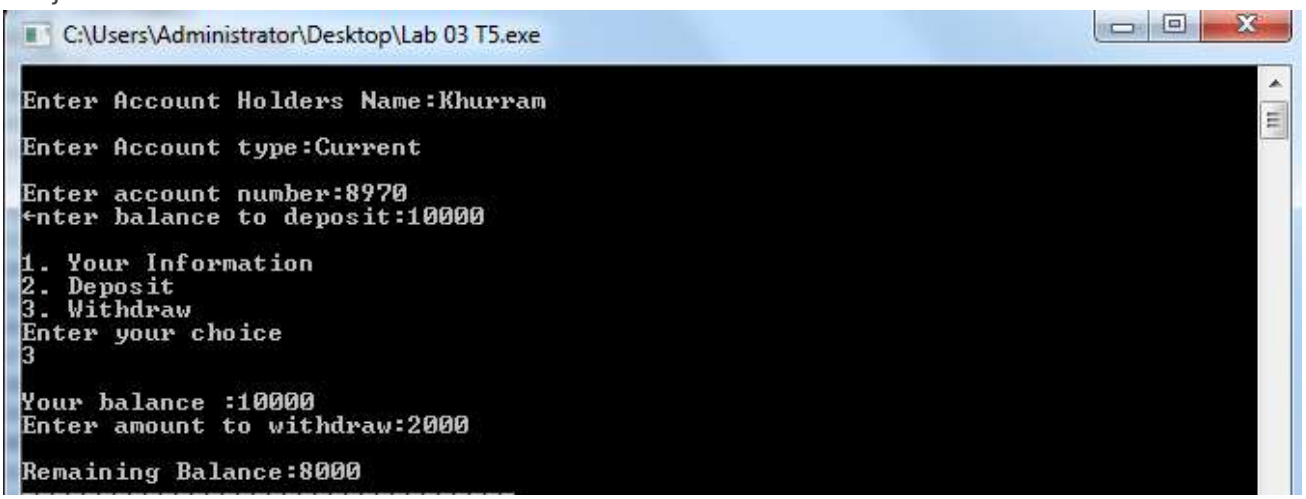
Code#5

```
#include<iostream>
using namespace std;
```

```
class Bank
{


public:
string name;
string account_type;
int account_number;
int balance;

//member functions of the class Bank
// initialize function to initialize data members

void initialize()
{
cout<<"\nEnter Account Holders Name:";
cin>>name;
cout<<"\nEnter Account type:";
cin>>account_type;
cout<<"\nEnter account number:";
cin>>account_number;
cout<<"\Enter balance to deposit:";
cin>>balance;
}


void deposit()
{
int bal;
cout<<"\nEnter the amout to deposit:";
cin>>bal;
balance+=bal;
cout<<"\nAmount deposited successfuly\nYour New Balance:"<<balance;
}

//check() function to withdraw amount and check remaining balance

void check()
{
int bal;
cout<<"\nYour balance :"<<balance<<"\nEnter amount to withdraw:";
```

```
cin>>bal;
if(bal<=balance)
{
balance-=bal;
cout<<"\nRemaining Balance:"<<balance;
}
else
{
exit(0);
}
}

//display function to display user information
void display()
{
cout<<"\nName :";
cin>>name;
cout<<"\nBalance :"<<balance;
}
};

int  main()
{
int i;


Bank bk;
bk.initialize();
cout<<"\n1. Your Information\n2. Deposit\n3. Withdraw\nEnter your choice\n";
cin>>i;
if(i==1)
{
bk.display();
}
else if(i==2)
{
bk.deposit();
}
else if(i==3)
{
bk.check();
}
```

```
return 0;
}
```


C:\Users\Administrator\Desktop\Lab 03 T5.exe
Enter Account Holders Name:Khurram
Enter Account type:Current
Enter account number:8970
←nter balance to deposit:10000
1. Your Information
2. Deposit
3. Withdraw
Enter your choice
3
Your balance :10000
Enter amount to withdraw:2000
Remaining Balance:8000

Object Oriented Programing (CL-1004)

Exercises

Task_01:

Create a class named "ShapeTriangle", your job is to print the area and perimeter of triangle, consider the triangle to have sides of 3, 4 and 5 units. The value of the area and perimeter needs to be printed by using a function

Task_02:

Create a class named "CompanyWorker", the class must contain the following appropriate functions, the goal is to print the final salary.

1 - A function named "InfoRetreival" which asks for the salary of the worker, the number of hours per work per day as arguments to the function.

2 - A function named "SalIncrease" which increases the salary of the employee by 5000PKR if it is below 25000.

3 - A function named "WorkBenefit" which increases the salary of the employee by 6000PKR if his/her work per day is more than 7 hours.

Task_03:

Create a class named "CustomerAccount", the class must contain the following attributes:

1) Name of the customer that is opening the account.
2) Preferred Account number (restrict it to 4 digits)
3) Either a Savings or current account.
4) The opening balance (must be >=5000)

The functions for this class should perform in the following manner

1) A function that provides the default values for the attributes.
2) A function that deposits the initial amount.
3) A function that checks the balance of the account and allows the customer to withdraw some amount (Not more than 25000 on a single transaction).
4) A function that prints the name and account balance of the customer.

Task_04:

Create a class named "ElectricBillCal", this class represents an electric company that charges it's consumers as per the following rates:

1) From unit 1 to 100, 5 PKR per unit
2) From unit 101 to 200, 7 PKR per unit
3) From unit 201 to 300, 9 PKR per unit
4) Anything above 300 11 PKR per unit

If the consumer uses as much units as his/her total bill exceeds 5000 PKR then an additional 13.5% duty must be charged.

You must create appropriate functions that notes and prints the names of the consumers, the units consumed by them and the overall cost incurred.

Task_05:

Create a class named "PhoneNumbers", the class should contain the following attributes:

1) STD
2) Number

You need to accept a valid number from at least 6 users and then rephrase the input number to a new changed number based on the following conditions:

1) The STD code should be added by 1 digit (Ex: 7 becomes 8)
2) Next you should reverse the first two digits of the numbers. (Ex: 823-38-985334 → 932-38-985334)

Print the changed numbers.

Task_06:

Create a class named "CarSpecs", this class must contain attributes namely doors, wheels, car_speed.

The default value for wheels is 4 and for doors 2 and speed is 0. You need to create two functions namely Civic and Rubicon. Civic should be set on default values while Rubicon to be incremented by 2 wheels and 2 doors. Two more functions namely Speed should increment the speed of vehicles by 7 while the break function should decrease the speed by 7. Print the current speed.