

OBJECT ORIENTED PROGRAMMING

OOP

PRACTICALS

NAME: SYEDA FAIZA ASLAM

SECTION: B

ROLL NO: CT-25064

LAB NO: 01

COURSE CODE: CT-260

COURSE TEACHER: Mr. AHMED ZAKI

Exercise

1. Write a program that take input of your roll number along with the marks obtained in five subjects and display the total marks obtained and the percentage.

SOURCE CODE:

```
// 1. Write a program that take input of your roll number along with the marks obtained in five  
// subjects and display the total marks obtained and the percentage.  
//================================================================  
=====  
  
#include<iostream>  
using namespace std;  
int main()  
{  
    int roll_no;  
    float marks[5];  
    float tot_marks=0.0;  
    float percentage;  
    cout<<"=====\n";  
    cout<<"Enter your Roll no: ";  
    cin>>roll_no;  
    cout<<"=====\n";  
    cout<<"Enter marks in 5 Subjects: \n";  
    cout<<"=====\n";  
    for(int i=0; i<5; i++)  
    {  
        cout<<"Marks in Subject "<<i+1<<": ";  
        cin>>marks[i];  
    }  
    for(int i=0; i<5; i++)
```

```
{  
    tot_marks+=marks[i];  
}  
cout<<"=====\n";  
cout<<"Total Marks: "<<tot_marks<<endl;  
percentage=(tot_marks/500)*100;  
cout<<"=====\n";  
cout<<"Percentage: "<<percentage<<"%"<<endl;  
cout<<"=====\n";  
return 0;  
}
```

OUTPUT:

```
C:\Users\AA\Desktop\OOP(PRACTICALS)\lab 1 q1.exe  
=====  
Enter your Roll no: 64  
=====  
Enter marks in 5 Subjects:  
=====  
Marks in Subject 1: 88  
Marks in Subject 2: 89  
Marks in Subject 3: 97  
Marks in Subject 4: 78  
Marks in Subject 5: 77  
=====  
Total Marks: 429  
=====  
Percentage: 85.8%  
=====  
-----  
Process exited after 21.08 seconds with return value 0  
Press any key to continue . . . -
```

2. Write a program to swap three numbers entered by a user using pointers.

SOURCE CODE:

```
//2. Write a program to swap three numbers entered by a user  
using pointers.  
//================================================================  
=====  
  
#include<iostream>  
using namespace std;  
int main()  
{  
    int n1,n2,n3;  
    int *a,*b,*c;  
    int x;  
    cout<<"Enter Three numbers for swapping: "<<endl;  
    cin>>n1;  
    cin>>n2;  
    cin>>n3;  
    a=&n1;  
    b=&n2;  
    c=&n3;  
    cout<<"=====\\n";  
    cout<<"BEFORE SWAPPING 3 INPUTED NUMBERS ARE: \\n";  
    cout<<"=====\\n";  
    cout<<"First Number: "<<*a<<endl;  
    cout<<"Second Number: "<<*b<<endl;  
    cout<<"Third Number: "<<*c<<endl;  
    x=*a;  
    *a=*b;
```

```
*b=*c;
*c=x;
cout<<"=====\n";
cout<<"AFTER SWAPPING 3 INPUTED NUMBERS ARE: \n";
cout<<"=====\n";
cout<<"First Number: "<<*a<<endl;
cout<<"Second Number: "<<*b<<endl;
cout<<"Third Number: "<<*c<<endl;

return 0;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\OOP(PRACTICALS)\lab 1 q2.exe
Enter Three numbers for swapping:
56
23
45
=====
BEFORE SWAPPING 3 INPUTED NUMBERS ARE:
=====
First Number: 56
Second Number: 23
Third Number: 45
=====
AFTER SWAPPING 3 INPUTED NUMBERS ARE:
=====
First Number: 23
Second Number: 45
Third Number: 56

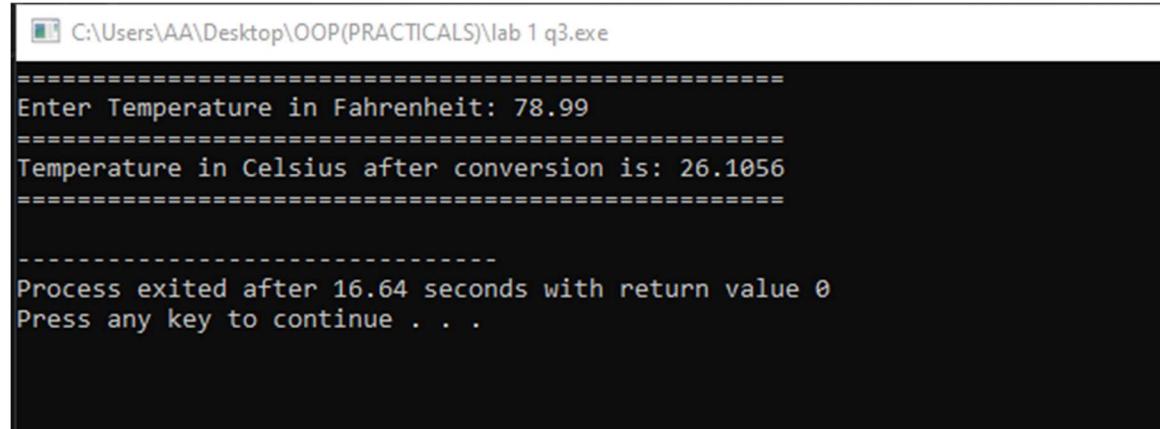
-----
Process exited after 4.821 seconds with return value 0
Press any key to continue . . .
```

3. Write a program to convert temp from Fahrenheit to Celsius unit using equation C=(F-32)/1.8

SOURCECODE:

```
//3. Write a program to convert temp from Fahrenheit to Celsius unit using  
equation C=(F-32)/1.8  
=====  
=====  
#include<iostream>  
using namespace std;  
int main()  
{  
    float templnF;  
    float templnC;  
    cout<<"=====\\n";  
    cout<<"Enter Temperature in Fahrenheit: ";  
    cin>>templnF;  
    templnC = (templnF-32)/1.8;  
    cout<<"=====\\n";  
    cout<<"Temperature in Celsius after conversion is: "<<templnC<<endl;  
    cout<<"=====\\n";  
    return 0;  
}
```

OUTPUT:



```
C:\Users\AA\Desktop\OOP(PRACTICALS)\lab 1 q3.exe  
=====  
Enter Temperature in Fahrenheit: 78.99  
=====  
Temperature in Celsius after conversion is: 26.1056  
=====  
-----  
Process exited after 16.64 seconds with return value 0  
Press any key to continue . . .
```

4. Using 2-D arrays, write a program that allows the user to input two, 3x3 matrices. Write a function for adding two matrices. Write another function for multiplying the two matrices.

SOURCECODE:

```
//4. Using 2-D arrays, write a program that allows the user to input two, 3x3  
matrices. Write a  
//function for adding two matrices. Write another function for multiplying the  
two matrices.  
//================================================================  
=====  
#include<iostream>  
using namespace std;  
  
void addMatrices(int arr1[3][3], int arr2[3][3]);  
void multiplyMatrices(int arr1[3][3], int arr2[3][3]);  
  
int main()  
{  
    int arr1[3][3];  
    int arr2[3][3];  
    cout<<"=====\n";  
    cout<<"Enter the values of Matrix A(3x3):\n";  
    cout<<"=====\n";  
    for(int i=0; i<3; i++)  
    {  
        for(int j=0; j<3; j++)  
        {  
            cout<<"Value at ["<<i+1<<"]["<<j+1<<"]: ";  
            cin>>arr1[i][j];  
        }  
    }  
    cout<<"The Matrix A will look like this: \n";
```

```

for(int i=0; i<3; i++)
{
    for(int j=0; j<3; j++)
    {
        cout<<"[ "<<arr1[i][j]<<" ]";
    }
    cout<<"\n";
}
cout<<"=====\n";
cout<<"Enter the Value for Matrix B(3x3):\n";
cout<<"=====\n";
for(int i=0; i<3; i++)
{
    for(int j=0; j<3; j++)
    {
        cout<<"Value at ["<<i+1<<"]["<<j+1<<"]: ";
        cin>>arr2[i][j];
    }
}
cout<<"The Matrix B will look like this: \n";
for(int i=0; i<3; i++)
{
    for(int j=0; j<3; j++)
    {
        cout<<"[ "<<arr2[i][j]<<" ]";
    }
    cout<<"\n";
}
cout<<"=====\n";
cout<<"ADDITION OF MATRIX A AND MATRIX B: \n";
cout<<"=====\n";
addMatrices(arr1, arr2);
cout<<"=====\n";
cout<<"MULTIPLICATION OF MATRIX A AND MATRIX B: \n";
cout<<"=====\n";

```

```

        multiplyMatrices(arr1, arr2);

        return 0;
    }
void addMatrices(int arr1[3][3], int arr2[3][3])
{
    int add[3][3];
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            add[i][j]=arr1[i][j]+arr2[i][j];
        }
    }
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            cout<<"["<<add[i][j]<<"]";
        }
        cout<<"\n";
    }
}
void multiplyMatrices(int arr1[3][3], int arr2[3][3])
{
    int mul[3][3]={0};
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            for(int k=0; k<3; k++)
            {
                mul[i][j]+=arr1[i][k]*arr2[k][j];
            }
        }
    }
}

```

```
    }
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            cout<<"["<<mul[i][j]<<"]";
        }
        cout<<"\n";
    }
}
```

OUTPUT:

```
C:\Users\AA\Desktop\OOP(PRACTICALS)\lab 1 q4.exe
=====
Value at [1][1]: 4
Value at [1][2]: 7
Value at [1][3]: 5
Value at [2][1]: 2
Value at [2][2]: 6
Value at [2][3]: 5
Value at [3][1]: 1
Value at [3][2]: 4
Value at [3][3]: 3
The Matrix B will look like this:
[ 4 ][ 7 ][ 5 ]
[ 2 ][ 6 ][ 5 ]
[ 1 ][ 4 ][ 3 ]
=====
ADDITION OF MATRIX A AND MATRIX B:
=====
[10][11][13]
[11][8][10]
[8][9][7]
=====
MULTIPLICATION OF MATRIX A AND MATRIX B:
=====
[40][98][74]
[45][95][70]
[42][95][72]

-----
Process exited after 73.65 seconds with return value 0
Press any key to continue . . . _
```

5. Write a program to find Surface area and volume of a sphere using functions.

SOURCECODE:

```
//5. Write a program to find Surface area and volume of a sphere using functions.  
//=====  
=====  
#include<iostream>  
using namespace std;  
void areaOfSphere(float r);  
void volOfSphere(float r);  
  
int main()  
{  
    float radius;  
    cout<<"=====\\n";  
    cout<<"Enter value of Radius for calculating Area of sphere & Volume of  
Sphere: ";  
    cin>>radius;  
    cout<<"=====\\n";  
    cout<<"AREA OF SPHERE: \\n";  
    cout<<"=====\\n";  
    areaOfSphere(radius);  
    cout<<"=====\\n";  
    cout<<"VOLUME OF SPHERE: \\n";  
    cout<<"=====\\n";  
    volOfSphere(radius);  
    cout<<"=====\\n";  
  
    return 0;  
}  
void areaOfSphere(float r)
```

```
{  
    float areaOfSphere;  
    areaOfSphere = (4.0)*(3.142)*(r*r);  
    cout<<"Area Of Sphere is: "<<areaOfSphere<<endl;  
}  
void volOfSphere(float r)  
{  
    float volOfSphere;  
    volOfSphere = (4.0/3.0)*(3.142)*(r*r*r);  
    cout<<"Volume of Sphere is: "<<volOfSphere<<endl;  
}
```

OUTPUT:

```
C:\Users\AA\Desktop\OOP(PRACTICALS)\lab 1 q5.exe  
=====  
Enter value of Radius for calculating Area of sphere & Volume of Sphere: 10.8  
=====  
AREA OF SPHERE:  
=====  
Area Of Sphere is: 1465.93  
=====  
VOLUME OF SPHERE:  
=====  
Volume of Sphere is: 5277.35  
=====  
-----  
Process exited after 12.41 seconds with return value 0  
Press any key to continue . . .
```

6. Write a program to help a bank create its withdrawal system. Your program should allow the user to input their account type. Account types are: savings, current. Following business rules apply when withdrawing from a certain account:

Savings:

User must provide the savings account number and code 'S' (for savings). When withdrawing from a savings account, users need to pay a set 2% of the money that they withdraw. If the amount of money withdrawn is over 50,000, then a 5% tax will be deducted. The money deducted shall be from the remaining money in the account.

Current:

User must provide the current account number and code „C“ (for current). When withdrawing from a current account, users need to pay a withdrawal fee of 100. If the amount of money withdrawn is over 50,000, then a 5% tax will be deducted. The money deducted shall be from the remaining money in the account.

Assume all users have the 200,000 in their accounts, and cannot withdraw more than 100,000 at a time.

SOURCECODE:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string acctype, accnum;
    char acccode;
    double withdrawamount;
```

```
double balance = 200000.0;
double fee = 0.0;
double tax = 0.0;
cout << "Enter Account Type (savings/current): ";
cin >> acctype;
cout << "Enter Account Number: ";
cin >> accnum;
cout << "Enter Account Code (S/C): ";
cin >> acccode;
cout << "Enter Amount to Withdraw: ";
cin >> withdrawamount;
if (withdrawamount > 100000)
{
    cout << "Error: Maximum withdrawal limit is 100,000." << endl;
}
if ((acctype == "savings" || acctype == "Savings") && acccode == 'S')
{
    fee = withdrawamount * 0.02;
    if (withdrawamount > 50000)
    {
        tax = withdrawamount * 0.05;
    }
}
else if ((acctype == "current" || acctype == "Current") && acccode == 'C')
{
    fee = 100;
    if (withdrawamount > 50000)
    {
        tax = withdrawamount * 0.05;
    }
}
else
{
    cout << "Invalid account type or code combination." << endl;
}
```

```
double totaldeduction = withdrawamount + fee + tax;

if (totaldeduction > balance)
{
    cout << "Insufficient amount to cover withdrawal and fees." << endl;
}
else
{
    balance -= totaldeduction;
    cout << " Withdrawal Successful" << endl;
    cout << "Amount: " << withdrawamount << endl;
    cout << "Fees: " << fee << endl;
    cout << "Tax: " << tax << endl;
    cout << "Remaining Balance: " << balance << endl;
}

return 0;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\lab 1 q6.exe
Enter Account Type (savings/current): savings
Enter Account Number: 321
Enter Account Code (S/C): S
Enter Amount to Withdraw: 560
Withdrawal Successful
Amount: 560
Fees: 11.2
Tax: 0
Remaining Balance: 199429

-----
Process exited after 17.33 seconds with return value 0
Press any key to continue . . .
```

EXAMPLE:

Example of a Basic C++ Program:

```
#include <iostream>
using namespace std;
int main() {
    char a;
    int num;
    cout << "Enter a character and an integer: ";
    cin >> a >> num;
    cout << "Character: " << a << endl;
    cout << "Number: " << num;
    return 0;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Enter a character and an integer: F
5
Character: F
Number: 5
-----
Process exited after 9.792 seconds with return value 0
Press any key to continue . . .
```

Example Code for Pointers:

```
#include <iostream>
using namespace std;
int main(){
    int *p;
    int x = 37;
    cout << "Line 1: x = " << x << endl; //Line 1
    p = &x; //Line 2
    //Line 3
    cout << "Line 3: *p = " << *p << ", x = " << x << endl;
    *p = 58; //Line 4
    //Line 5
    cout << "Line 5: *p = " << *p << ", x = " << x << endl;
    cout << "Line 6: Address of p = " << &p << endl; //Line 6
    cout << "Line 7: Value of p = " << p << endl; //Line 7
    cout << "Line 8: Value of the memory location " << "pointed to
    by *p = " << *p << endl; //Line 8
    cout << "Line 9: Address of x = " << &x << endl; //Line 9
    cout << "Line 10: Value of x = " << x << endl; //Line 10
    return 0;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Line 1: x = 37
Line 3: *p = 37, x = 37
Line 5: *p = 58, x = 58
Line 6: Address of p = 0x78fe18
Line 7: Value of p = 0x78fe14
Line 8: Value of the memory location pointed to by *p = 58
Line 9: Address of x = 0x78fe14
Line 10: Value of x = 58

-----
Process exited after 0.3128 seconds with return value 0
Press any key to continue . . .
```

Example Code:

```
#include<iostream>
using namespace std;
int main(){
```

```
int* intPtr;
char* charArray;
int Size;
intPtr = new int; // allocating memory to single variable
cout << "Enter an Integer Value: ";
cin >> *intPtr;
cout << "Enter the size of the Character Array : ";
cin >> Size;
charArray = new char[Size];//allocating memory to array
for (int i = 0; i < Size; i++)
    cin >> charArray[i];
for (int i = 0; i < Size; i++)
    cout << charArray[i];
return 0;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Enter an Integer Value: 3
Enter the size of the Character Array : 5
1
2
4
5
3
12453
-----
Process exited after 15.11 seconds with return value 0
Press any key to continue . . .
```

Example Code:

```
#include <iostream>
using namespace std;
void funcValueParam (int num);
int main (){
    int number = 6; //Line 1
    cout << "Line 2: Before calling the function " <<
    "funcValueParam, number = " << number << endl; //Line 2
    funcValueParam(number); //Line 3
    cout << "Line 4: After calling the function "      <<
    "funcValueParam, number = " << number << endl; //Line 4
```

```
        return 0;
    }
void funcValueParam (int num){
    cout << "Line 5: In the function funcValueParam, " <<
    "before changing, num = " << num << endl; //Line 5
    num = 15; //Line 6
    cout << "Line 7: In the function funcValueParam, " <<
    "after changing, num = " << num << endl; //Line 7
}
```

OUTPUT:

```
[1] C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Line 2: Before calling the function funcValueParam, number = 6
Line 5: In the function funcValueParam, before changing, num = 6
Line 7: In the function funcValueParam, after changing, num = 15
Line 4: After calling the function funcValueParam, number = 6

-----
Process exited after 0.6851 seconds with return value 0
Press any key to continue . . .
```

Example Code:

```
//This program reads a course score and prints the
//associated course grade.
#include <iostream>
using namespace std;
void getScore (int& score);
void printGrade (int score);
int main (){
    int courseScore;
    cout << "Line 1: Based on the course score, \n" << "this
        program computes the " << "course grade." << endl; //Line 1
    getScore(courseScore); //Line 2
    printGrade(courseScore); //Line 3
    return 0; }
void getScore (int& score){
    cout << "Line 4: Enter course score: "; //Line 4
    cin >> score; //Line 5
    cout << endl << "Line 6: Course score is " << score << endl;
    //Line 6
}
void printGrade (int cScore){
    cout << "Line 7: Your grade for the course is "; //Line 7
    if (cScore >= 90) //Line 8
        cout << "A." << endl;
    else if (cScore >= 80)
        cout << "B." << endl;
    else if (cScore >= 70)
        cout << "C." << endl;
    else if (cScore >= 60)
        cout << "D." << endl;
    else
        cout << "F." << endl;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Line 1: Based on the course score,
this program computes the course grade.
Line 4: Enter course score: 88

Line 6: Course score is 88
Line 7: Your grade for the course is B.

-----
Process exited after 2.169 seconds with return value 0
Press any key to continue . . .
```

Example Code for Static Variables:

```
//Program: Static and automatic variables
#include <iostream>
using namespace std;
void test ();
int main (){
    int count;
    for (count = 1; count <= 5; count++)
        test ();
    return 0;
}
void test (){
    static int x = 0;
    int y = 10;
    x = x + 2;
    y = y + 1;
    cout << "Inside test x = " << x << " and y = " << y << endl;
}
```

OUTPUT:

```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
Inside test x = 2 and y = 11
Inside test x = 4 and y = 11
Inside test x = 6 and y = 11
Inside test x = 8 and y = 11
Inside test x = 10 and y = 11
-----
Process exited after 0.3033 seconds with return value 0
Press any key to continue . . .
```

