

OBJECT ORIENTED PROGRAMMING (LAB 1)

EXERCISES

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.

```
4 #include <iostream>
5 using namespace std;
6 int main(){
7     int roll_no;
8     float marks[5], total =0, percentage;
9
10    cout<<"Enter roll number: ";
11    cin>>roll_no;
12
13    for(int i=0; i<5; i++){
14        cout<<"Enter marks of subject "<<i+1<<": ";
15        cin>>marks[i];
16        total += marks[i];
17    }
18    percentage = (total / 500) *100;
19
20    cout<<"Roll Number: "<<roll_no<<endl;
21    cout<<"Total Makrs: "<<total<<endl;
22    cout<<"Percetnage: "<<percentage<<endl;
23    return 0;
24 }
```

```
C:\Users\PC\Desktop\All Folders\OOP-LAB\Lab-1\Ques-1.exe

Enter roll number: 70
Enter marks of subject 1: 89
Enter marks of subject 2: 78
Enter marks of subject 3: 90
Enter marks of subject 4: 90
Enter marks of subject 5: 99
Roll Number: 70
Total Makrs: 446
Percetnage: 89.2

-----
Process exited after 11.79 seconds with return value 0
Press any key to continue . . .
```

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

```
2 #include <iostream>
3 using namespace std;
4 void swapNumbers(int *a, int *b, int *c)
5 {
6     int temp;
7     temp = *a;
8     *a = *b;
9     *b = *c;
10    *c = temp;
11 }
12 int main()
13 {
14     int num1, num2, num3;
15
16     cout << "Enter first number: ";
17     cin >> num1;
18     cout << "Enter second number: ";
19     cin >> num2;
20     cout << "Enter third number: ";
21     cin >> num3;
22     swapNumbers(&num1, &num2, &num3);
23     cout << "After swapping:" << endl;
24     cout << "First number: " << num1 << endl;
25     cout << "Second number: " << num2 << endl;
26     cout << "Third number: " << num3 << endl;
27
28 }
```

```
C:\Users\PC\Desktop\All Folders\OOP-LAB\Lab-1\Ques-2.exe
Enter first number: 12
Enter second number: 9
Enter third number: 8
After swapping:
First number: 9
Second number: 8
Third number: 12

-----
Process exited after 3.876 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

```
2
3 #include <iostream>
4 using namespace std;
5 int main(){
6     float Fahrenheit;
7     float Celsius;
8
9     cout<<"Enter temp in Fahrenheit: ";
10    cin>>Fahrenheit;
11
12    Celsius = (Fahrenheit-32)/1.8;
13
14    cout<<"Temperature in Fahrenheit is = "<<Fahrenheit<<endl;
15    cout<<"Temperature in Celsius is = "<<Celsius<<endl;
16
17    return 0;
18
19 }
```

```
C:\Users\PC\Desktop\All Folders\OOP-LAB\Lab-1\Ques-3.exe
Enter temp in Fahrenheit: 200
Temperature in Fahrenheit is = 200
Temperature in Celsius is = 93.3333

-----
Process exited after 2.312 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.

```
1 #include <iostream>
2 using namespace std;
3 // Function to add two matrices
4 void addMatrix(int A[3][3], int B[3][3], int sum[3][3]) {
5     for(int i = 0; i < 3; i++) {
6         for(int j = 0; j < 3; j++) {
7             sum[i][j] = A[i][j] + B[i][j];
8         }
9     }
10 }
11 // Function to multiply two matrices
12 void multiplyMatrix(int A[3][3], int B[3][3], int mul[3][3]) {
13     for(int i = 0; i < 3; i++) {
14         for(int j = 0; j < 3; j++) {
15             mul[i][j] = 0;
16             for(int k = 0; k < 3; k++) {
17                 mul[i][j] += A[i][k] * B[k][j];
18             }
19         }
20     }
21 }
22 // Function to display a matrix
23 void displayMatrix(int M[3][3]) {
24     for(int i = 0; i < 3; i++) {
25         for(int j = 0; j < 3; j++) {
26             cout << M[i][j] << " ";
27         }
28         cout << endl;
29     }
30 }
31 int main(){
32     int A[3][3], B[3][3];
33     int sum[3][3], mul[3][3];
34
35     cout << "Enter elements of 1st matrix:\n";
36     for(int i = 0; i < 3; i++)
37         for(int j = 0; j < 3; j++)
38             cin >> A[i][j];
39
40     cout << "Enter elements of 2nd matrix:\n";
41     for(int i = 0; i < 3; i++)
42         for(int j = 0; j < 3; j++)
43             cin >> B[i][j];
44
45     addMatrix(A, B, sum);
46     multiplyMatrix(A, B, mul);
47
48     cout << "\nMatrix A:\n";
49     displayMatrix(A);
50
51     cout << "\nMatrix B:\n";
52     displayMatrix(B);
53
54     cout << "\nAddition of matrices:\n";
55     displayMatrix(sum);
56
57     cout << "\nMultiplication of matrices:\n";
58     displayMatrix(mul);
59
60     return 0;
61 }
62 }
```

```
C:\Users\PC\Desktop>All Folders\OOP-LAB\Lab-1\Ques-4.exe
4 5 6
12 3 4
Enter elements of 2nd matrix:
5 4 3
8 9 0
7 0 6

Matrix A:
1 2 3
4 5 6
12 3 4

Matrix B:
5 4 3
8 9 0
7 0 6

Addition of matrices:
6 6 6
12 14 6
19 3 10

Multiplication of matrices:
42 22 21
102 61 48
112 75 60
```

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

```
4
5 #include <iostream>
6 using namespace std;
7
8 float surfaceArea(float r){
9     return 4 * 3.142 * r * r;
10 }
11
12 float volume(float r){
13     return (4.0 / 3.0) * 3.1416 * r * r * r;
14 }
15
16 int main(){
17     float radius;
18     cout<<"Enter radius: ";
19     cin>>radius;
20
21     cout<<"Surface Area of sphere is: "<<surfaceArea(radius)<<endl;
22     cout<<"Volume of sphere is: "<<volume(radius)<<endl;
23
24     return 0;
25 }
26 }
```

```
C:\Users\PC\Desktop>All Folders\OOP-LAB\Lab-1\Ques-5.exe
Enter radius: 4
Surface Area of sphere is: 201.088
Volume of sphere is: 268.083

-----
Process exited after 3.071 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.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int balance = 200000;
5     char accountType;
6     int accountNumber;
7     int withdrawAmount;
8
9     cout<<"Enter account type ('s' for saving 'c' for current): ";
10    cin>>accountType;
11
12    cout << "Enter account number: ";
13    cin >> accountNumber;
14
15    cout << "Enter withdrawal amount: ";
16    cin >> withdrawAmount;
17
18    if(withdrawAmount > 100000){
19        cout<<"you cant withdraw more than 100000";
20    }
21    if (withdrawAmount > balance) {
22        cout << "Insufficient balance.";
23    }
24
25    if(accountType == 's'){
26        balance = balance - withdrawAmount;
27
28        int fee = withdrawAmount * 0.02;
29        balance = balance - fee;
30
31        if (withdrawAmount > 50000) {
32            int tax = balance * 0.05;
33            balance = balance - tax;
34        }
35
36
37        cout << "Savings account withdrawal successful.\n";
38        cout << "Remaining balance: " << balance;
39    }
40
41    // Current Account
42    else if (accountType == 'C') {
43        balance = balance - withdrawAmount;
44        balance = balance - 100;
45        if (withdrawAmount > 50000) {
46            int tax = balance * 0.05;
47            balance = balance - tax;
48        }
49        cout << "Current account withdrawal successful.\n";
50        cout << "Remaining balance: " << balance;
51    }
52
53    // Invalid Account Type
54    else {
55        cout << "Invalid account type.";
56    }
57
58    return 0;
59 }
```

```
C:\Users\PC\Desktop>All Folders\OOP-LAB\Lab-1\Ques-6.exe
Enter account type ('s' for saving 'c' for current): s
Enter account number: 1234
Enter withdrawal amount: 2000
Savings account withdrawal successful.
Remaining balance: 197960
-----
Process exited after 16.24 seconds with return value 0
Press any key to continue . . .
```

EXAMPLES

Example 1 and output:

```
#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;
}
```

```
Enter a character and an integer: a 6
Character: a
Number: 6
```

Example 2 and output:

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string greeting = "Hello";
    cout << greeting;
    return 0;
}
```

```
Hello
```

Example 3 and output:

```
#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;
}
```

```
Line 1: x = 37
Line 3: *p = 37, x = 37
Line 5: *p = 58, x = 58
Line 6: Address of p = 0x6ffe08
Line 7: Value of p = 0x6ffe04
Line 8: Value of memory location pointed to by *p = 58
Line 9: Address of x = 0x6ffe04
Line 10: Value of x = 58
```

Example 4 and output:

```
#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;
}
```

```
Enter an Integer Value: 3
Enter the size of the Character Array : 3
w e t
wet
```

Example 5 and output

```
#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
}
```

```
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
```

Example 6 and output:

```
//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;
}
```

```
Line 1: Based on the course score,
this program computes the course grade.
Line 4: Enter course score: 87

Line 6: Course score is 87
Line 7: Your grade for the course is B
```

Example 7 and output:

```
//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;
}
```

```
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
```