

## Lab#7

### ➤ Display your name on the condole:



### ➤ C++ Student information:

```
name.cpp > main0
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      string name="Iqra";
6      cout<<"Name :"<<name<<endl;
7      int studentID = 123456;
8      cout<<"StudentID"<<studentID<<endl;
9      string department = "Computer science";
10     cout<<"department"<<department<<endl;
11     int age = 19;
12     cout<<"age"<<age<<endl;
13     float gpa = 3.7;
14     cout<<"gpa"<<gpa<<endl;
15
16     return 0;
17 }
18
19
20
```

## Lab#8

### ➤ Question#1

**# Step 2: Initialize** numbers

```
number1 = 10
```

```
number2 = 20
```

**# Step 3: Calculate the sum**

```
answer = number1 + number2
```

**# Step 4: Print the answer**

When you run this program, it will output:

**The sum is: 30**

### ➤ Question#2

**# Step 2: Prompt the user for the first number**

```
print("Enter the first number:")
```

```
number1 = float(input()) # Convert input to float for handling decimals
```

**# Step 4: Prompt the user for the second number**

```
print("Enter the second number:")
```

```
number2 = float(input())
```

**# Step 6: Calculate the sum**

```
sum = number1 + number2
```

### **# Step 7: Print the result**

```
print("The sum is:", sum)
```

Example Output:

Enter the first number:

10

Enter the second number:

20

**The sum is: 30.0**

### **➤ Question#3**

### **# Step 2: Prompt the user for the value in USD**

```
print("Enter value in USD:")
```

```
usd = float(input()) # Convert input to float to handle decimals
```

### **# Step 4: Convert USD to PKR**

```
pkr = usd * 170
```

### **# Step 5: Print the value in PKR**

```
print("The value in PKR is:", pkr)
```

### **Enter value in USD:**

10

**The value in PKR is: 1700.0**

➤ **Question#4**

Expression:

$$(a * (b + c)) + (c * (a + c))$$

---

Memory Representation Steps:

1. Reserve memory cells:

MA1 for a

MA2 for b

MA3 for c

2. Perform the computation step by step:

Step 1: Compute  $b + c$  and store in MA2 (overwriting b).

Step 2: Compute  $a * (b + c)$  using MA1 and MA2, store the result in MA1.

Step 3: Compute  $a + c$  and store in MA2 (overwriting  $b + c$ ).

Step 4: Compute  $c * (a + c)$  using MA3 and MA2, store the result in MA3.

Step 5: Add the results from MA1 and MA3, and store the final result in MA1.

### 3. Output the result stored in MA1

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // Reserve memory cells
6      int MA1, MA2, MA3;
7
8      // Input values for a, b, and c
9      cout << "Enter the value of a: ";
10     cin >> MA1; // MA1 represents 'a'
11
12     cout << "Enter the value of b: ";
13     cin >> MA2; // MA2 represents 'b'
14
15     cout << "Enter the value of c: ";
16     cin >> MA3; // MA3 represents 'c'
17
18     // Step 1: Compute b + c and store in MA2
19     MA2 = MA2 + MA3; // MA2 now contains (b + c)
20
21     // Step 2: Compute a * (b + c) and store in MA1
22     MA1 = MA1 * MA2; // MA1 now contains a * (b + c)
23
24     // Step 3: Compute a + c and store in MA2
25     MA2 = (MA1 / MA2) + MA3; // MA2 now contains (a + c)
26
27     // Step 4: Compute c * (a + c) and store in MA3
28     MA3 = MA3 * MA2; // MA3 now contains c * (a + c)
29
30     // Step 5: Add results from MA1 and MA3, and store in MA1
31     MA1 = MA1 + MA3; // MA1 now contains the final result
32
33     // Output the result
34     cout << "The result is: " << MA1 << endl;
35
36     return 0;
37 }
```

Input:

Enter the value of a: 2

Enter the value of b: 3

Enter the value of c: 4

Output:

The result is: 34

### ➤ Problem#5

Calculate the Area of a Rectangle

Formula:

$$\text{Area} = \text{Length} \times \text{Width}$$

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      float length, width, area;
6
7      // Input length and width
8      cout << "Enter the length of the fence in feet: ";
9      cin >> length;
10
11     cout << "Enter the width of the fence in feet: ";
12     cin >> width;
13
14     // Calculate area
15     area = length * width;
16
17     // Display the result
18     cout << "The area of the fence is: " << area << " square feet" << endl;
19
20     return 0;
21 }
```

Act  
Go to



### ➤ Problem#6

Calculate Final Velocity of a Toy Car

Formula for acceleration:

$$\text{Acceleration} = \frac{\text{Final Velocity} - \text{Initial Velocity}}{\text{Time}}$$

$$\text{Final Velocity} = (\text{Acceleration} \times \text{Time}) + \text{Initial Velocity}$$

```
3
4  int main() {
5      float initialVelocity, acceleration, time, finalVelocity;
6
7      // Input initial velocity, acceleration, and time
8      cout << "Enter the initial velocity of the car (m/s): ";
9      cin >> initialVelocity;
10
11      cout << "Enter the acceleration of the car (m/s^2): ";
12      cin >> acceleration;
13
14      cout << "Enter the time (s): ";
15      cin >> time;
16
17      // Calculate final velocity
18      finalVelocity = (acceleration * time) + initialVelocity;
19
20      // Display the result
21      cout << "The final velocity of the car is: " << finalVelocity << " m/s" << endl;
22
23      return 0;
24 }
```



## ➤ Problem#7

Calculate Marks Percentage

Total marks for 5 subjects = 500.

Formula for percentage:

$$\text{Percentage} = \frac{\text{Total Obtained Marks}}{\text{Total Marks}} \times 100$$

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string studentName;
6      float marks[5], total = 0, percentage;
7
8      // Input student name
9      cout << "Enter your name: ";
10     cin >> studentName;
11
12     // Input marks for 5 subjects
13     cout << "Enter marks for 5 subjects:" << endl;
14     for (int i = 0; i < 5; i++) {
15         cout << "Enter subject " << i + 1 << " marks: ";
16         cin >> marks[i];
17         total += marks[i];
18     }
19
20     // Calculate percentage
21     percentage = (total / 500) * 100;
22
23     // Display result
24     cout << "Student Name: " << studentName << endl;
25     cout << "Total Marks: " << total << "/500" << endl;
26     cout << "Percentage: " << percentage << "%" << endl;
27
28     return 0;
29 }
```

### ➤ Problem#8

perform Operations on 15 Numbers

1. Add the first 5 numbers.
2. Multiply the next 5 numbers.
3. Subtract the last 5 numbers.
4. Add the results of steps 1 and 2, then subtract the result of step 3.

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

int main() {
    int numbers[15];
    int sum1 = 0, product2 = 1, sum3 = 0, finalResult;

    // Input 15 numbers
    cout << "Enter 15 numbers:" << endl;
    for (int i = 0; i < 15; i++) {
        cin >> numbers[i];
    }

    // Step 1: Add the first 5 numbers
    for (int i = 0; i < 5; i++) {
        sum1 += numbers[i];
    }

    // Step 2: Multiply the next 5 numbers
    for (int i = 5; i < 10; i++) {
        product2 *= numbers[i];
    }

    // Step 3: Subtract the last 5 numbers
    for (int i = 10; i < 15; i++) {
        sum3 += numbers[i];
    }

    // Step 4: Final calculation
    finalResult = sum1 + product2 - sum3;

    // Display the result
    cout << "The final result is: " << finalResult << endl;

    return 0;
}
```

## ➤ Problem#9

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int number, sum = 0;
6
7      // Input the number
8      cout << "Enter a number: ";
9      cin >> number;
10
11     // Calculate the sum of the digits
12     while (number > 0) {
13         sum += number % 10; // Extract the last digit and add it to sum
14         number /= 10;       // Remove the last digit
15     }
16
17     // Display the result
18     cout << "The sum of the digits is: " << sum << endl;
19
20     return 0;
21 }
```

Explanation of Test Cases

Input: 1234

1. Extract digits: 4, 3, 2, 1.
2. Sum:  $4 + 3 + 2 + 1 = 10$ .
3. Output: 10.

Input: 4324

1. Extract digits: 4, 2, 3, 4.
2. Sum:  $4 + 2 + 3 + 4 = 13$ .
3. Output: 13.

Input: 4901

1. Extract digits: 1, 0, 9, 4.
2. Sum:  $1 + 0 + 9 + 4 = 14$ .



3. Output: 14.

### >Problem#10

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int number, reverse = 0;
6
7      // Input the number
8      cout << "Enter a number: ";
9      cin >> number;
10
11     // Calculate the reverse of the number
12     while (number > 0) {
13         int digit = number % 10;    // Extract the last digit
14         reverse = reverse * 10 + digit; // Append it to the reversed number
15         number /= 10;               // Remove the last digit
16     }
17
18     // Display the reversed number
19     cout << "The reversed number is: " << reverse << endl;
20
21     return 0;
22 }
```

Explanation of Test Cases

Input: 1234

1. Extract digits: 4, 3, 2, 1.
2. Reverse: 4, 43, 432, 4321.
3. Output: 4321.

Input: 4324

1. Extract digits: 4, 2, 3, 4

2. Reverse: 4, 42, 423, 4234.

3. Output: 4234.

Input: 4901

1. Extract digits: 1, 0, 9, 4.

2. Reverse: 1, 10, 109, 1094.

3. Output: 1094

## Lab#9

### Task#1

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int num;
6
7      // Input the number
8      cout << "Enter an integer: ";
9      cin >> num;
10
11     // Check whether the number is even or odd
12     if (num % 2 == 0) {
13         cout << num << " is an even number." << endl;
14     } else {
15         cout << num << " is an odd number." << endl;
16     }
17
18     return 0;
19 }
```

Test Cases

Input: 4

1.  $4 \% 2 = 0 \rightarrow$  Even.

2. Output: 4 is an even number.

Input: 7

1.  $7 \% 2 = 1 \rightarrow$  Odd.

2. Output: 7 is an odd number.

Input: 0

1.  $0 \% 2 = 0 \rightarrow$  Even

2. Output 0 is an even number.

## Task#2

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int number, reverse = 0;
6
7      // Input the number
8      cout << "Enter a number: ";
9      cin >> number;
10
11     // Calculate the reverse of the number
12     while (number > 0) {
13         int digit = number % 10;    // Extract the last digit
14         reverse = reverse * 10 + digit; // Append it to the reversed number
15         number /= 10;               // Remove the last digit
16     }
17
18     // Display the reversed number
19     cout << "The reversed number is: " << reverse << endl;
20
21     return 0;
22 }
```

Test Cases

Input: 5 and 10



1. Compare:  $5 < 10$
2. Output: 10 is larger.

Input: 20 and 15

1. Compare:  $20 > 15$ .
2. Output: 20 is larger

Input: 8 and 8

1. Compare: Both are equal.

### Task#3

```
1  #include <iostream>
2  #include <string> // For string handling
3  using namespace std;
4
5  int main() {
6      string word1, word2;
7
8      // Input two words
9      cout << "Enter the first word: ";
10     cin >> word1;
11     cout << "Enter the second word: ";
12     cin >> word2;
13
14     // Check if the words are the same using comparison operator ==
15     if (word1 == word2) {
16         cout << "Yes, the words are the same." << endl;
17     } else {
18         cout << "No, the words are different." << endl;
19     }
20
21     return 0;
22 }
```

Test Cases

Input 1:

Enter the first word: Ali



Enter the second word: Al

Output:

Yes, the words are the same.

Input 2:

Enter the first word: Ali

Enter the second word: ali

Output:

No, the words are different.

Input 3:

Enter the first word: Hello

Enter the second word: World

Output:

No, the words are different.

## Task#5

```
1  # Take the first number as input
2  num1 = float(input("Enter the first number: "))
3
4  # Take the operator as input
5  operator = input("Enter the operator (+, -, *, /, %): ")
6
7  # Take the second number as input
8  num2 = float(input("Enter the second number: "))
9
10 # Perform the operation based on the operator
11 if operator == "+":
12     result = num1 + num2
13 elif operator == "-":
14     result = num1 - num2
15 elif operator == "*":
16     result = num1 * num2
17 elif operator == "/":
18     # Handle division by zero
19     if num2 != 0:
20         result = num1 / num2
21     else:
22         result = "Error! Division by zero."
23 elif operator == "%":
24     # Handle modulus by zero
25     if num2 != 0:
26         result = num1 % num2
27     else:
28         result = "Error! Modulus by zero."
29 else:
30     result = "Invalid operator!"
31
32 # Print the result
33 print("Result:", result)
```