

# C++ MODEL EXAM:

## 1.sum of elements in array

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     float fahrenheit, celsius;
7
8     cout << "Enter the temperature in Celsius : ";
9     cin >> celsius;
10    fahrenheit = (celsius * 9.0) / 5.0 + 32;
11    cout << "The temperature in Celsius : " << celsius << endl;
12    cout << "The temperature in Fahrenheit : " << fahrenheit << endl;
13    return 0;
14 }
```

```
C:\Users\avalakunta sai\Docu x + v -
Enter the temperature in Celsius : 4
The temperature in Celsius : 4
The temperature in Fahrenheit : 39.2

-----
Process exited after 2.752 seconds with return val
Press any key to continue . . . |
```

## 2.celsius and fahrenheit

```
Untitled1.cpp Untitled2.cpp Untitled3.cpp Untitled4.cpp Untitled5.cpp Untitled6.cpp
1 #include <iostream>
2 using namespace std;
3
4 int sum(int arr[], int n)
5 {
6     int sum = 0;
7     for (int i = 0; i < n; i++)
8         sum += arr[i];
9
10    return sum;
11 }
12
13 int main()
14 {
15     int arr[] = { 1,2,3,4};
16     int n = sizeof(arr) / sizeof(arr[0]);
17     cout << "Sum of given array is " << sum(arr, n);
18     return 0;
19 }
```

```
C:\Users\avalakunta sai\Docu x + v -
Sum of given array is 10
-----
Process exited after 0.08635 seconds with return value 0
Press any key to continue . . . |
```

## 3.pattern

```
Untitled1.cpp Untitled2.cpp Untitled3.cpp Untitled4.cpp Untitled5.cpp
#include <iostream>
using namespace std;

int main() {

    int rows;

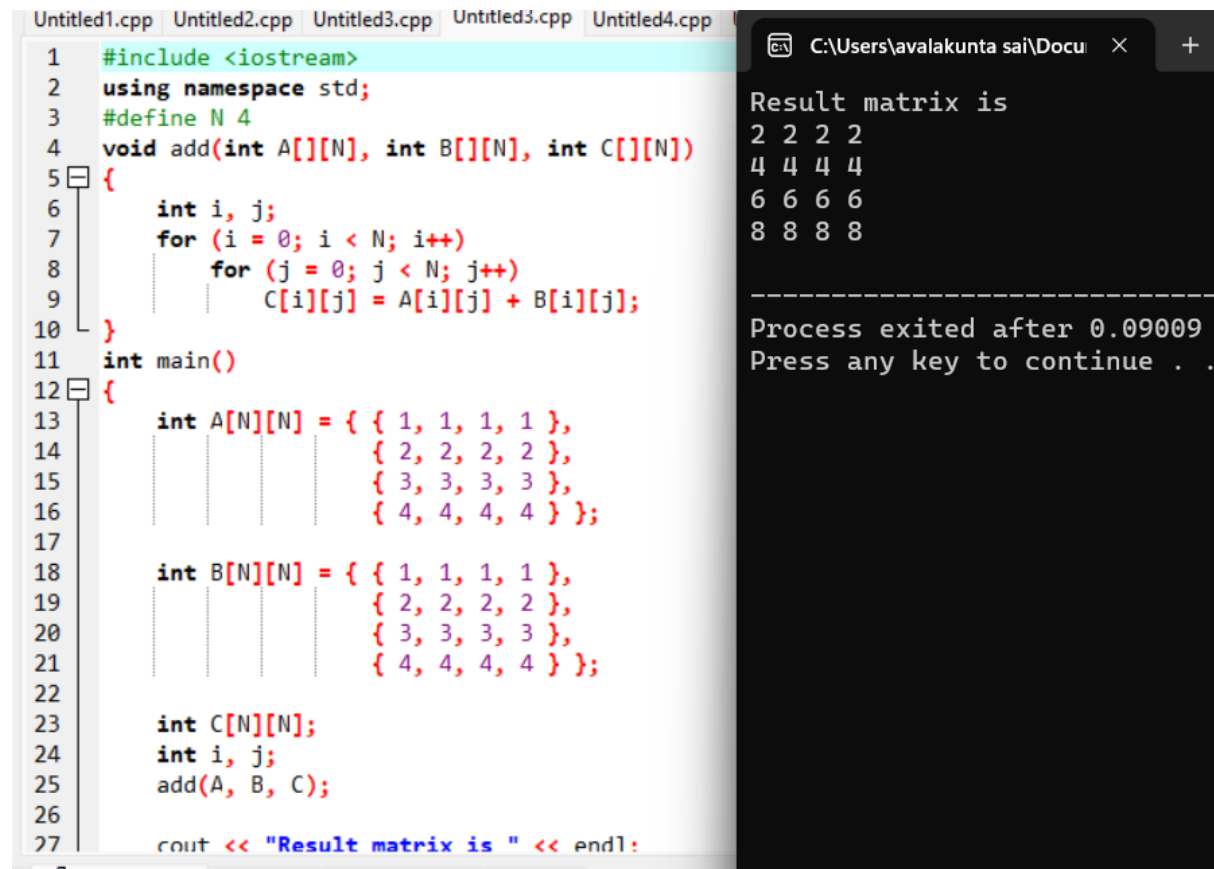
    cout << "Enter number of rows: ";
    cin >> rows;

    for(int i = 1; i <= rows; ++i) {
        for(int j = 1; j <= i; ++j) {
            cout << j << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

```
Result matrix is
2 2 2 2
4 4 4 4
6 6 6 6
8 8 8 8

-----
Process exited after 0.08142 s
Press any key to continue . . . |
```

#### 4.add 2 matrix



The image shows a C++ IDE with a code editor on the left and a terminal window on the right. The code editor displays a C++ program for adding two 4x4 matrices. The program defines a constant N=4, a function add() that takes three 2D arrays and adds the second to the first, and a main() function that initializes two matrices A and B, calls add(A, B, C), and prints the result matrix C. The terminal window shows the output of the program, which is the result matrix C, followed by a message indicating the process exited after 0.09009 seconds and a prompt to press any key to continue.

```
1 #include <iostream>
2 using namespace std;
3 #define N 4
4 void add(int A[][N], int B[][N], int C[][N])
5 {
6     int i, j;
7     for (i = 0; i < N; i++)
8         for (j = 0; j < N; j++)
9             C[i][j] = A[i][j] + B[i][j];
10 }
11 int main()
12 {
13     int A[N][N] = { { 1, 1, 1, 1 },
14                     { 2, 2, 2, 2 },
15                     { 3, 3, 3, 3 },
16                     { 4, 4, 4, 4 } };
17
18     int B[N][N] = { { 1, 1, 1, 1 },
19                     { 2, 2, 2, 2 },
20                     { 3, 3, 3, 3 },
21                     { 4, 4, 4, 4 } };
22
23     int C[N][N];
24     int i, j;
25     add(A, B, C);
26
27     cout << "Result matrix is " << endl;
```

```
Result matrix is
2 2 2 2
4 4 4 4
6 6 6 6
8 8 8 8

-----
Process exited after 0.09009
Press any key to continue . .
```

#### 5.exception handling.

```
Untitled1.cpp  Untitled2.cpp  Untitled3.cpp  Untitled3.cpp  Untitled4.cpp  Untitled5.cpp  Untitled6.cpp

int main() {

    double numerator, denominator, divide;

    cout << "Enter numerator: ";
    cin >> numerator;

    cout << "Enter denominator: ";
    cin >> denominator;

    try {

        if (denominator == 0)
            throw 0;

        divide = numerator / denominator;
        cout << numerator << " / " << denominator << " = " << divide;

    }

    catch (int num_exception) {
        cout << "Error: Cannot divide by " << num_exception << endl;
    }

    return 0;
}
```

Enter numerator: 4  
Enter denominator: 56  
4 / 56 = 0.0714286  
-----  
Process exited after 3.3 seconds with return value 0  
Press any key to continue . . .

## 6.constructor and destructor.

```
Untitled1.cpp  Untitled2.cpp  Untitled3.cpp  Untitled3.cpp  Untitled4.cpp  Untitled5.cpp  Untitled6.cpp  Untitled7.cpp

1  #include<iostream>
2  #include<stdio.h>
3  #include<string.h>
4
5  using namespace std;
6
7  class T4Tutorials
8  {
9      int acno;
10     char AccountHolderName[100], Account_Type[100];
11     float bal;
12 public:
13     T4Tutorials(int acc_no, char *name, char *acc_type, float Balance)
14     {
15         acno=acc_no;
16         strcpy(AccountHolderName, name);
17         strcpy(Account_Type, acc_type);
18         bal=Balance;
19     }
20     void deposit();
21     void withdraw();
22     void Show();
23
24 };
25 void T4Tutorials::deposit()
26 {
27     float DepositAmount;
28     cout << "Enter Deposit Amount: ";
29     cin >> DepositAmount;
30     bal = bal + DepositAmount;
31     Show();
32 }
```

Enter Details:  
\*\*\*\*\*  
Account No.  
123444455  
Name :  
sai  
Account Type :  
savings  
Balance :  
10000  
Enter Deposit Amount =  
10000000  
Enter Withdraw Amount =  
2334  
\*\*\*\*\*  
Account No. : 123444455 Name : sai Account Type : savings Balance : 1.00077e+007  
-----  
Process exited after 24.5 seconds with return value 0  
Press any key to continue . . .

## 7.squareroot.

```

1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 int main()
5 {
6     cout<<"squre root of 25=";
7     cout<<sqrt(25);
8     return 0;
9 }

```

squre root of 25=5

-----  
 Process exited after 0.0829 seconds with return value 0  
 Press any key to continue . . . |

8.create a base class for employee.

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class Employee {
5 protected:
6     string Emp_name;
7     int Emp_id;
8     string Address;
9     string Mail_id;
10    string Mobile_no;
11 public:
12    Employee(const string& name, int id, const string& address, const string& mail, const string& mobile)
13        : Emp_name(name), Emp_id(id), Address(address), Mail_id(mail), Mobile_no(mobile) {}
14    virtual void generatePaySlip() = 0;
15 };
16 class Programmer : public Employee {
17 protected:
18     double BP;
19 public:
20    Programmer(const string& name, int id, const string& address, const string& mail, const string& mobile, double BP)
21        : Employee(name, id, address, mail, mobile), BP(BP) {}
22    void generatePaySlip() override {
23        double DA = 0.97 * BP;
24        double HRA = 0.10 * BP;
25        double PF = 0.12 * BP;
26        double staffClubFund = 0.001 * BP;
27    }
28 };

```

Pay Slip for Programmer  
 Employee Name: John Doe  
 Employee ID: 101  
 Address: 123 Street Ave  
 Mail ID: john@example.com  
 Mobile Number: 1234567890  
 Basic Pay: 50000  
 Dearness Allowance: 48500  
 House Rent Allowance: 5000  
 Provident Fund: 6000  
 Staff Club Fund: 50  
 Gross Salary: 103500  
 Net Salary: 97450

-----  
 Process exited after 0.08865 seconds with return value 0  
 Press any key to continue . . . |