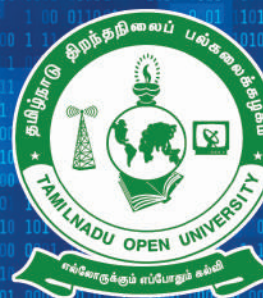
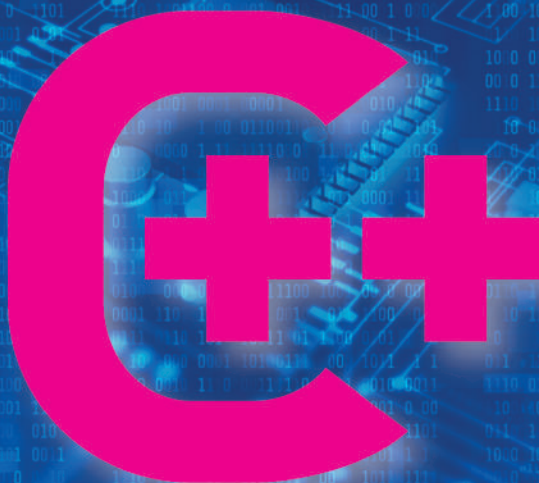


BCAS-P3



தமிழ்நாடு திறந்தநிலைப் பல்கலைக்கழகம்  
எல்லோருக்கும் எப்போதும் கல்வி

B.C.A., Second Year  
SEMESTER - 03



LABORATORY  
MANUAL

SCHOOL OF COMPUTER SCIENCE

TAMIL NADU OPEN UNIVERSITY

577, ANNA SALAI, SAIDAPET, CHENNAI - 600 015





**தமிழ்நாடு திறந்தநிலைப் பல்கலைக்கழகம்**

**Bachelor of Computer Applications (B.C.A.)**

**[SECOND YEAR – THIRD SEMESTER]**

**BCAS – P3**

**C++ PROGRAMMING LABORATORY MANUAL**

**Department of Computer Science**

**School of Computer Science**

**Tamil Nadu Open University**

**577, Anna Salai, Saidapet, Chennai – 600015**

**July, 2022**

<b>Name of Programme</b>	<b>Bachelor of Computer Applications (BCA)</b>
<b>Name of the Course</b>	<b>C++ Laboratory</b>
<b>Course Code</b>	<b>BCAS – P3</b>
<b>Curriculum Design</b>	<b>Dr. R. KALAIARASI</b> Assistant Professor School of Computer Science Tamil Nadu Open University, Chennai. & <b>Er. N. SIVASHANMUGAM</b> Assistant Professor School of Computer Science Tamil Nadu Open University, Chennai.
<b>Course Writer</b>	<b>Dr. R. HEMAMALINI</b> Assistant Professor Sakthi Engineering College Thiruninravur (D.t)
<b>Content Editor &amp; Course Coordinator</b>	<b>Dr. R. KALAIARASI</b> Assistant Professor School of Computer Science Tamil Nadu Open University, Chennai

**July, 2022 (First Edition)**

**ISBN No: 978-93-95841-68-9**

© Tamil Nadu Open University

*All rights reserved. No part of this work can be reproduced in any form, by mimeograph or any other means, without permission in writing from the Tamil Nadu Open University. Course writer is the sole responsible person for the contents presented/available in the Course material.*

*Further information on the Tamil Nadu Open University Academic Programmes may be obtained from the University Office at 577, Anna Salai, Saidapet, Chennai - 600 015, Tamilnadu, India [or] [www.tnou.ac.in](http://www.tnou.ac.in)*



@TNOU, 2022, "C++ Laboratory" is made available under a Creative Commons Attribution-Share Alike 4.0 License (International)  
<http://creativecommons.org/licenses/by-sa/4.0>

*Printed by:*

*M/s. R.K. 11, Virudhunagar District Co-operative Printing Press Ltd, Virudhunagar – 626003.*



# TAMIL NADU OPEN UNIVERSITY

(A State Open University Established by Government of Tamil Nadu, Recognized by UGC & DEB,  
Member in Asian Association of Open Universities & Association of Commonwealth Universities)

No.577, Anna Salai, Saidapet, Chennai - 600 015. Tamil Nadu.

04.07.2022

## Message

My Dear Beloved Learner,

At the outset, I am pleased to invite you to the Tamil Nadu Open University, a leading Open and Distance Education Institution in India. I am doubly happy to know that you have joined a Programme in the University for upgrading your educational qualification. The University has been striving hard to democratize the tertiary education to all sections of the society with a singular motto "education to anyone at anytime" since its inception in 2002. You can feel proud that you are a learner of the University that has got the UGC's approval vide F.No.1-6/2018 (DEB) dated 03.10.2018 for offering 81 UG and PG Programmes through distance mode. As you are aware, the Government of Tamilnadu vide G.O. (MS) No.107 assured that those, who have secured a Degree in the Tamil Nadu Open University through 10+2+3 Pattern are duly eligible for government jobs.

With dedicated and uncompromising learner supporting services, the University facilitates the pertinent learning needs such as (a) socially relevant updated curriculum, (b) resourceful Self-Learning Materials prepared by the eminent academics, (c) conduction of induction programmes for the first year learners, (d) arranging counseling classes and examination centres within your vicinity, (e) reliable grievance redressal mechanism and so on. In order to provide equal education on par with on campus education, the University holds its Learner Support Centres (LSCs) at 91 Government Arts and Science Colleges across the State, as per the approval of the Government of Tamilnadu.

At present, the University operates eight Regional Centres each one in Chennai, Coimbatore, Dharmapuri, Madurai, Tirunelveli, The Nilgiris, Tiruchirappalli and Vilupuram. These Regional Centres facilitate the robust learner supporting services. Learners can get their essential services at the respective Regional Centre rather than approaching the Head Quarters. Moreover, our University Website holds a separate page for "Student Portal", in which you can get almost all the details relating to your Programme with just clicking the mouse. To reduce the time and cost, the Online Admission facility is also extended to those who are inclined to join in any Programme being offered by the University from their places.

At this juncture, I assure that the Tamil Nadu Open University will always extend its constant academic supports from admission to placement effectively.

You can get more details at **[www.tnou.ac.in](http://www.tnou.ac.in)** or **044-24306600**.

I wish you the best of luck in all of your future endeavours!

**Principal Secretary**  
Higher Education Department



Email: [registrartnou@gmail.com](mailto:registrartnou@gmail.com)



Ph: 044-24306645  
Fax: 91-44-24306640



URL: [www.tnou.ac.in](http://www.tnou.ac.in)

## **SYLLABUS**

**COURSE TITLE : C++ Programming Laboratory**  
**COURSE CODE : BCAS-P3**  
**COURSE CREDIT : 02**

### **COURSE OBJECTIVES**

While studying the C++ Laboratory course, the student shall be able to:

- Know about how to write basic C++ programming
- Learn how to use control statements
- Understand the usage of arrays and functions
- Practice how to create classes

### **List of Programmes**

1. Write a C++ program to calculate average marks scored by a student for 3 subjects.
2. Write a C++ program to find the area and perimeter of a circle and rectangle.
3. Write a C++ program to swap two numbers.
4. Write a C++ program to find largest of three numbers.
5. Write a C++ program to find the maximum number among three numbers.
6. Write a C++ program to generate Fibonacci series.
7. Write a C++ program to perform string manipulation.
8. Find the length of a string. Compare two strings, Concatenate two strings, Reverse a string, Copy a string to another location.
9. Write a C++ program to find quotient and remainder of 2 numbers.
10. Write a C++ program to manipulate the class account using classes and function. A user should be able to perform the following functions. Deposit money. Withdraw money, Calculate the interest Check the total balance in his account.
11. Write a C++ program to generate Prime numbers between 1 and 50.
12. Write a C++ program to perform matrix addition and multiplication.
13. Write a C++ program to check whether the given matrix is a sparse matrix or not.
14. Write a C++ program to overload unary minus operator.
15. Write a C++ program to calculate total sales and average sales made by a salesman.

### **COURSE OUTCOMES**

After completion of the C++ Laboratory course, the student can be able to:

- Understand how to write programs in C++
- Practice how to use control statements
- Learn about arrays and functions
- Know how to create classes

### **SCHEME OF LESSONS**

<b>S. No.</b>	<b>Contents</b>	<b>Page No.</b>
<b>1.</b>	C++ program to calculate average marks scored by a student for 3 subjects.	<b>1</b>
<b>2.</b>	Program to find the area and perimeter of a circle and rectangle.	<b>3</b>
<b>3.</b>	Program to swap two numbers.	<b>6</b>
<b>4.</b>	Program to find largest of three numbers.	<b>7</b>
<b>5.</b>	Program to find the maximum number among three numbers.	<b>11</b>
<b>6.</b>	Program to generate Fibonacci series.	<b>15</b>
<b>7.</b>	Program to perform string manipulation.	<b>17</b>
<b>8.</b>	Find the length of a string. Compare two strings, Concatenate two strings, Reverse a string, Copy a string to another location.	<b>22</b>
<b>9.</b>	Program to find quotient and remainder of 2 numbers.	<b>30</b>
<b>10.</b>	C++ program to manipulate the class account using classes and function. A user should be able to perform the following functions. Deposit money. Withdraw money, Calculate the interest Check the total balance in his account.	<b>32</b>
<b>11.</b>	Program to generate Prime numbers between 1 and 50.	<b>37</b>
<b>12.</b>	Program to perform matrix addition and multiplication.	<b>38</b>
<b>13.</b>	Program to check whether the given matrix is a sparse matrix or not.	<b>41</b>
<b>14.</b>	Program to overload unary minus operator.	<b>43</b>
<b>15.</b>	Program to calculate total sales and average sales made by a salesman.	<b>45</b>

## **INSTRUCTION TO THE CANDIDATES**

### **General Instructions to run programming language in C:**

- Open Dev-C++ or any other programming C code editor platform.
- Include Header files or library files.
- Open **void main()** function.
- We can clear the data from the console by using **clrscr()** function.
- Perform **conditional and unconditional statements** for respective programs.
- Every statement should concluded with **semicolon(;;)**.
- To display the message by using **printf()** function.
- To get the input from the user by using **Scanf()** function.
- The function is used to read the character from the keyboard given by the user using **getch()** function.
- To terminate the execution of a function and returns the control to the calling function using the **return** statement.
- Extension for C file to save is “**.c**”. (Ex :filename.c)
- Finally press **F9** to compile the program and press **F10** to execute the program.

---

**1. Write a C++ program to calculate average marks scored by a student for 3 subjects.**

---

**Aim: To calculate average marks scored by a student for 3 subjects.**

Assume n1, n2, and n3 are the marks of three subject scored by a student. Then the total marks can be calculated as (n1+n2+n3) and average mark can be calculated as (total marks/3).

Total Mark = n1+n2+n3

Average Mark = total\_mark / 3

**Algorithm:**

1. Declare the variables.
2. Declare the input values.
3. Calculate the Total Marks
4. Calculate the Average Marks
5. Display the Result.

**Program:**

```
#include<iostream.h>
using namespace std;
int main()
{
    // declare variables
    int n1, n2, n3;
    int total;
    double avg;

    // take input for marks of 3 sub
    cout << "Enter Marks of 3 Subjects: ";
    cin >> n1 >> n2 >> n3;

    // calculate total marks
```



```
total = n1 + n2 + n3;

// calculate average mark
avg = total/3;

// display result
cout << "Total Mark = " << total << endl;
cout << "Average Mark = " << avg << endl;

return 0;
}
```

**Output:**

Enter Marks of 3 Subjects: 75 81 84

Total Mark = 240

Average Mark = 80

---

## 2. Write a C++ program to find the area and perimeter of a circle and rectangle.

---

**Aim:** To find the area and perimeter of a circle and rectangle.

**Algorithm:**

1. Declare the variables.
2. Declare the input values for length and width of the rectangle.
3. Calculate the Area and Perimeter.
4. Display the Result.

**Program**

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

int main()
{
    int width, lngth, area, peri;
    cout << "\n\n Find the Area and Perimeter of a Rectangle :\n";
    cout << "-----\n";
    cout << "Input the length of the rectangle : ";
    cin >> lngth;
    cout << "Input the width of the rectangle : ";
    cin >> width;
    area = (lngth * width);
    peri = 2 * (lngth + width);
    cout << "The area of the rectangle is : " << area << endl;
    cout << "The perimeter of the rectangle is : " << peri << endl;

    cout << endl;
    return 0;
}
```

### Output:

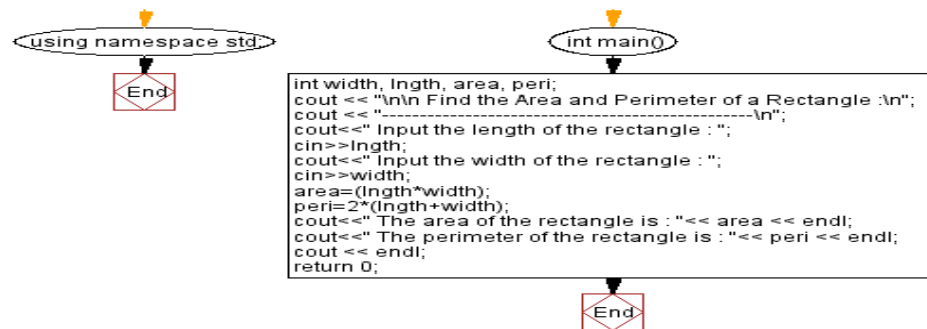
Input the length of the rectangle : 10

Input the width of the rectangle : 15

The area of the rectangle is : 150

The perimeter of the rectangle is : 5

### Flow chart:



### Algorithm:

1. Declare the variables.
2. Declare the input values for length and width of the circle.
3. Calculate the Area and circumference.
4. Display the Result.

### Program

```
#include <iostream.h>
```

```
#define PI 3.14159
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float radius, area, circum;
```

```
    cout << "\n\n Find the area and circumference of any circle :\n";
```

```

    cout << "-----\n";
    cout<<" Input the radius(1/2 of diameter) of a circle : ";
    cin>>radius;

    circum = 2*PI*radius;
    area = PI*(radius*radius);

    cout<<" The area of the circle is : "<< area << endl;
    cout<<" The circumference of the circle is : "<< circum << endl;

    cout << endl;
    return 0;
}

```

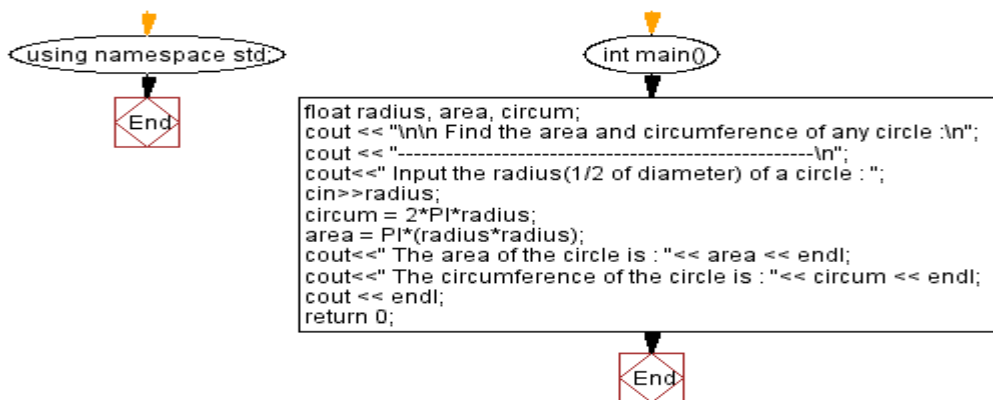
### **Output:**

Input the radius (1/2 of diameter) of a circle: 5

The area of the circle is: 78.5397

The circumference of the circle is: 31.4159

### **Flowchart:**





---

### 3. Write a C++ program to swap two numbers.

---

**Aim:** To swap two numbers.

**Algorithm:**

1. Declare the input values, temporary variable.
2. Display keyword "Before swapping".
3. Display the values.
4. Display keyword "After swapping".
5. Display the values after swapping.

**Program:**

```
#include <iostream.h>
using namespace std;
int main()
{
    int a = 5, b = 10, temp;
    cout << "Before swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;
    temp = a;
    a = b;
    b = temp;
    cout << "\nAfter swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;
    return 0;
}
```

**Output:**

Before swapping:

a = 5, b = 10

After swapping:

a = 10, b = 5

---

#### **4. Write a C++ program to find largest of three numbers.**

---

**Aim:** To find largest of three numbers.

This program can be done using three ways.

**Algorithm:**

1. Ask the user to enter three float values.
2. Read the three float values in num1, num2, and num3 (integer variables).
3. Check if num1 is greater than num2.
4. If true, then check if num1 is greater than num3.  
    If true, then print 'num1' as the greatest number.  
    If false, then print 'num3' as the greatest number.
5. If false, then check if num2 is greater than num3.  
    1. If true, then print 'num2' as the greatest number.  
    2. If false, then print 'num3' as the greatest number.

**Case 1: if statement**

```
#include <iostream.h>

using namespace std;

int main()
{
    float n1, n2, n3;

    cout << "Enter three numbers: ";

    cin >> n1 >> n2 >> n3;

    if(n1 >= n2 && n1 >= n3)
        cout << "Largest number: " << n1;

    if(n2 >= n1 && n2 >= n3)
        cout << "Largest number: " << n2;

    if(n3 >= n1 && n3 >= n2)
        cout << "Largest number: " << n3;
```

```
    return 0;  
}
```

**Output:**

Enter three numbers: 2.3

8.3

-4.2

Largest number: 8.3

**Case 2: if else statement**

**Algorithm:**

1. Ask the user to enter three float values.
2. Read the three float values in num1, num2, and num3 (integer variables).
3. Check if num1 is greater than num2 or equal to num2.
4. If true, then check if num1 is greater than num3 or equal to num3.  
    If true, then print 'num1' as the greatest number.  
    If false, then print 'num3' as the greatest number.
5. If false, then check if num2 is greater than num1 or equal to num1.
6. If True, then check if num2 is greater than num3 or equal to num3.
  1. If true, then print 'num2' as the greatest number.
  2. If false, then print 'num3' as the greatest number.

**Program:**

```
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    float n1, n2, n3;  
    cout << "Enter three numbers: ";
```

```

cin >> n1 >> n2 >> n3;
if((n1 >= n2) && (n1 >= n3))
    cout << "Largest number: " << n1;
else if ((n2 >= n1) && (n2 >= n3))
    cout << "Largest number: " << n2;
else
    cout << "Largest number: " << n3;
return 0;
}

```

### **Output:**

Enter three numbers: 2.3

8.3

-4.2

Largest number: 8.3

### **Case 3: if else if statement**

#### **Algorithm:**

1. Ask the user to enter three float values.
2. Read the three float values in num1, num2, and num3 (integer variables).
3. Check if num1 is greater than num2 or equal to num2.
4. If true, then check if num1 is greater than num3 or equal to num3.
  - If true, then print 'num1' as the greatest number.
  - If false, then print 'num3' as the greatest number.
5. If false, then check if num2 is greater than num1 or equal to num1.
6. If True, then check if num2 is greater than num3 or equal to num3.
  1. If true, then print 'num2' as the greatest number.
  2. If false, then print 'num3' as the greatest number.



**Program:**

```
#include <iostream.h>

using namespace std;

int main()
{
    float n1, n2, n3;

    cout << "Enter three numbers: ";

    cin >> n1 >> n2 >> n3;

    if (n1 >= n2)
    {
        if (n1 >= n3)
            cout << "Largest number: " << n1;
        else
            cout << "Largest number: " << n3;
    }
    else

        if (n2 >= n3)
        {
            cout << "Largest number: " << n2;
            else
                cout << "Largest number: " << n3;
        }

    return 0;
}
```

**Output:**

Enter three numbers: 2.3

8.3

-4.2

Largest number: 8.3

---

## 5. Write a C++ program to find maximum among three numbers.

---

**Aim:** To find maximum among three numbers.

This program can be done using three ways.

**Algorithm:**

1. Ask the user to enter three integer values.
3. Check if 'a' is greater than b and 'a' is greater than c.
4. If true, then print 'a' is the maximum number.
5. If false, then check if 'b' is greater than 'a' and 'b' is greater than 'c'.
  1. If true, then print 'b' is the maximum number.
6. If false, then check if 'c' is greater than 'a' and 'c' is greater than 'b'.
  1. If true, then print 'c' as the maximum number.

### **Case 1: if statement**

```
#include <iostream.h>
using namespace std;
int main()
{
    int a = 11;
    int b = 5;
    int c = 32;
    int max;
    if (a > b && a > c)
    {
        max = a;
    }
    if (b > a && b > c)
    {
```

```

        max = b;
    }
    if (c > a && c > b)
    {
        max = c;
    }

    cout << max << endl;
return 0;
}

```

**Output:**

32

**Case 2: if else if statement**

**Algorithm:**

1. Enter three integer values.
3. Check if 'a' is greater than b and 'a' is greater than c.
4. If true, then print 'a' is the maximum number.
5. If false, then check if 'b' is greater than 'C'.
  1. If true, then print 'b' is the maximum number.
6. If false, then print 'c' as the maximum number.

**Program:**

```

#include <iostream>

using namespace std;

int main() {
    int a = 11;
    int b = 55;
    int c = 23;
    int max;

```

```

    if (a>b && a>c)
    {
        max = a;
    }
else if (b>c)
{
    max = b;
}
else
{
    max = c;
}

cout << max << endl;
return 0;
}

```

**Output:**

32

**Case 3: Nested if else statement**

**Algorithm:**

1. Enter the three integer values.
3. Check if 'a' is greater than 'b'.
4. If true, then check if 'a' is greater than 'c'.
  1. If true, then print 'a' is the maximum number.
  2. If false, then print 'c' is the maximum number.
5. If false, then check if 'b' is greater than 'c'.
  1. If true, then print 'b' is the maximum number.
  2. If false, then print 'c' is the maximum number.



**Program:**

```
#include <iostream.h>
using namespace std;
int main() {
    int a = 11;
    int b = 5;
    int c = 2;
    int max;
    if (a > b)
    {
        if (a > c)
        {
            max = a;
        }
    }
    else
    {
        max = c;
    }
    else
    {
        if (b > c)
        {
            max = b;
        }
    }
    else
    {
        max = c;
    }
    cout << max << endl;
    return 0;
}
```

**Output:**

32

---

## 6. Write a C++ program to generate Fibonacci series.

---

**Aim:** To generate Fibonacci series.

**Algorithm:**

1. Declare variables i, t1, t2, nextTerm.
2. Initialize the variables, t1=0, t2=1, and nextTerm =0
3. Enter the number of terms of Fibonacci series to be printed
4. Check the condition. Print first two terms of series
5. Use loop for the following steps
  - > nextTerm=t1+t2
  - > t1=t2
  - > t2=nextTerm
  - > Increase value of i each time by 1
  - > print the value of nextTerm.

**Program:**

```
#include <iostream.h>

using namespace std;

int main()
{
    int n, t1 = 0, t2 = 1, nextTerm = 0;
    cout << "Enter the number of terms: ";
    cin >> n;
    cout << "Fibonacci Series: ";
    for (int i = 1; i <= n; ++i)
    {
        // Prints the first two terms.
        if(i == 1)
        {
```

```

        cout << t1 << ", ";
        continue;
    }
    if(i == 2)
    {
        cout << t2 << ", ";
        continue;
    }
    nextTerm = t1 + t2;
    t1 = t2;
    t2 = nextTerm;
    cout << nextTerm << ", ";
}
return 0;}

```

**Output:**

Enter the number of terms: 10

Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

---

## **7. Write a C++ program to perform string manipulation.**

---

**Aim:** To perform string manipulation.

**C++ String:**

**Aim:** TO DISPAY STRING

**Algorithm:**

1. Start
2. Declare the strings
3. Display the strings as output.

**Program:**

```
#include <iostream.h>

using namespace std;

int main( ) {

    string s1 = "Hello";

    char ch[] = { 'C', '+', '+' };

    string s2 = string(ch);

    cout<<s1<<endl;

    cout<<s2<<endl;

}
```

**Output:**

Hello

C++

**C++ String Compare:**

**Aim:** To compare string

**Algorithm:**

1. key ='Mango'.
2. Declare the buffer size.

3. Do check the condition.
4. Input the buffer.
5. If the condition satisfies.
6. Print the result.

**Program:**

```
#include <iostream>

#include <cstring>

using namespace std;

int main ()
{
    char key[] = "mango";
    char buffer[50];
    do {
        cout<<"What is my favourite fruit? ";
        cin>>buffer;
    } while (strcmp (key,buffer) != 0);
    cout<<"Answer is correct!!"<<endl;
    return 0;
}
```

**Output:**

```
What is my favourite fruit? Apple
What is my favourite fruit? Banana
What is my favourite fruit? Mango
Answer is correct.
```

## **C++ String Concat:**

### **Aim: To Concat string**

### **Algorithm:**

1. Initialize char Key and Buffer.
2. Input the key string and buffer string values.
3. Concat the key and buffer.
4. Display the Result.

### **Program:**

```
#include <iostream.h>
#include <cstring.h>
using namespace std;
int main()
{
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    cout << "Enter the buffer string: ";
    cin.getline(buffer, 25);
    strcat(key, buffer);
    cout << "Key = " << key << endl;
    cout << "Buffer = " << buffer << endl;
    return 0;
}
```

### **Output:**

Enter the key string: Welcome to  
Enter the buffer string: C++ Programming.  
Key = Welcome to C++ Programming.



Buffer = C++ Programming.

### **C++ String Copy:**

**Aim:** To perform String copy.

### **Algorithm:**

1. Initialize char Key and Buffer.
2. Input the key string value.
3. Copy the key string to buffer string.
4. Display the Result.

### **Program:**

```
#include <iostream.h>
#include <cstring.h>
using namespace std;
int main()
{
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    strcpy(buffer, key);
    cout << "Key = " << key << endl;
    cout << "Buffer = " << buffer << endl;
    return 0;
}
```

### **Output:**

Enter the Key string: C++ Language

Key = C++ Tutorial

Buffer = C++ Tutorial

### **C++ String Length:**

**Aim:** To find the length

### **Algorithm:**

1. Start
2. Enter the character array.
3. Calculate the length of the string and Display the output.
4. Otherwise return 0.

### **Program:**

```
#include <iostream.h>
#include <cstring.h>
using namespace std;
int main()
{
    char ary[] = "Welcome to C++ Programming";
    cout << "Length of String = " << strlen(ary)<<endl;
    return 0;
}
```

### **Output:**

Length of String = 26

---

**8. Write a C++ program to find the length of the string, compare two strings, Concatenate two strings, Reverse a string, Copy a string to another location.**

---

**Aim :** To find the length of the string

**Algorithm:**

1. Start
2. Enter the character array.
3. Calculate the length of the string and Display the output.
4. Otherwise return 0.

**Find the length of the string:**

```
#include <iostream.h>

using namespace std;

int main() {

    string str = "C++ Programming";

    // you can also use str.length()

    cout << "String Length = " << str.size();

    return 0;

}
```

**Output:**

String Length = 15

**Compare Two Strings without strcmp()**

**Aim:** To compare two strings without using the strcmp( )

**Algorithm**

1. Input two character strings.
2. Check that both the first value of the strings does not have null value.
3. Compare the strings each value of the first string with the second string.

4. If the check value (chk) is equal to zero then both the strings are equal else they are different.

5. Stop execution.

**Program:**

```
#include<iostream.h>

using namespace std;

int main()
{
    char str1[50], str2[50];
    int i=0, chk=0;
    cout<<"Enter the First String: ";
    cin>>str1;
    cout<<"Enter the Second String: ";
    cin>>str2;
    while(str1[i]!='\0' || str2[i]!='\0')
    {
        if(str1[i]!=str2[i])
        {
            chk = 1;
            break;
        }
        i++;
    }
    if(chk==0)
        cout<<"\nStrings are Equal";
    else
        cout<<"\nStrings are not Equal";
    cout<<endl;
```

```
    return 0;
}
```

**Output:**

Enter the First String: computer

Enter the Second String: computer

Strings are equal.

**Compare Two Strings using strcmp()**

**Aim:** To Compare Two Strings using strcmp().

**Algorithm:**

1. Input two character strings.
2. Check that both the first value of the strings does not have null value.
3. The function, strcmp() takes two string as argument and returns 0 if both strings are equal. And the function strlen() takes single string as argument and returns its length.
4. Stop execution

**Program:**

```
#include<iostream.h>

#include<string.h>

using namespace std;

int main()
{
    char str1[50], str2[50];

    int len1, len2;

    cout<<"Enter the First String: ";

    cin>>str1;

    cout<<"Enter the Second String: ";

    cin>>str2;
```

```

len1 = strlen(str1);
len2 = strlen(str2);
if(len1==len2)
{
    if(strcmp(str1, str2)==0)
        cout<<"\nStrings are Equal";
    else
        cout<<"\nStrings are not Equal";
}
else
    cout<<"\nStrings are not Equal";
cout<<endl;
return 0;
}

```

**Output:**

Enter the First String: computer  
Enter the Second String: computer  
Strings are equal.

**Concatenate Two Strings:**

**Aim:** To Concatenate Two Strings

**Algorithm:**

1. Input two character strings.
2. There are two character arrays in the above syntax, arr1 and arr2, which passed inside the strcat() function to return a concatenated string.
3. Stop Execution.

### **Program:**

**strcat() function:** The strcat is an inbuilt function of the string class, which adds two character strings to return a concatenated string.

### **Syntax:**

```
strcat ( char *arr1, char *arr2)
```

### **Program:**

```
#include <bits/stdc++.h>

#include <string.h>

using namespace std;

int main()

{

    // declare and initialize the string

    char str1[] = " We love";

    char str2[] = " C++ Programming";

    cout << " String 1: " <<str1 <<endl;

    cout << " String 2: " <<str2 <<endl;

    // use the strcat() function to concatenate the string

    strcat(str1, str2);

    cout << " The concatenated string is: " <<str1;

    return 0;

}
```

### **Output:**

String 1: We love

String 2: C++ Programming

The concatenated string is: We love C++ Programming.

### **Reverse a String:**

**Aim:** To Reverse a String

### **Algorithm:**

1. Input a string.
2. Create a loop to get the characters in the string one by one .and store it in another temp location.
3. Now copy the temp location into the reversed string location using a loop to count one by one.
4. The inputed string is reversed and stored in the required place.
5. Stop Execution.

### **PROGRAM:**

```
#include<iostream.h>

#include<string.h>

using namespace std;

int main ()

{

    char str[50], temp;

    int i, j;

    cout << "Enter a string : ";

    gets(str);

    j = strlen(str) - 1;

    for (i = 0; i < j; i++,j--)

    {
```



```

        temp = str[i];

        str[i] = str[j];

        str[j] = temp;

    }

    cout << "\nReverse string : " << str;

    return 0;

}

```

### **Output:**

Enter a string: MATHS

Reverse string: SHTAM

### **Copy a String from one to another:**

**Aim:** To copy the contents from one string into another string.

### **Algorithm:**

1. Input one string.
2. Declare two memory's for two strings source and destination.
3. str2: str2 is the destination string object which retains the copied string
4. The source string is copied to the destination string using string copy function.
5. Stop Execution.

This function copies the contents of one string into another string.

### **Syntax:**

Suppose str1 and str2 are two string objects, len is the length of substring. We want to copy string str1 into the string object str2 then the syntax would look like:

```
str1.copy(str2,len);
```

```
str1.copy(str2,len,pos);
```

**Parameter:**

str2: str2 is the destination string object which retains the copied string.

len: It defines the length of substring.

pos: It determines the position of first character to include.

**Return value:**

It returns the number of characters to be copied.

**Program:**

```
#include<iostream.h>

using namespace std;

int main()

{

    string source = " C++ PROGRAMMING ";

    char destination[20];

    cout<<"source string is : "<<source<<"\n";

    source.copy(destination,sizeof source);

    cout<<"destination string is : "<<destination;

    return 0; }
```

**Output:**

source string is: C++ PROGRAMMING

destination string is: C++ PROGRAMMING

---

### 9. Write a C++ program to find quotient and remainder of 2 numbers.

---

**Aim:** To find quotient and remainder of 2 numbers.

**ALGORITHM:**

1. Start the program.
2. Declare the variables divisor, dividend, quotient, and remainder.
3. Get the input values for dividend and divisor.
4. Calculate the quotient and remainder using the formula.
5. Display the output.
6. Stop the program.

**PROGRAM:**

```
#include <iostream.h>

using namespace std;

int main()
{
    int divisor, dividend, quotient, remainder;

    cout << "Enter dividend: ";

    cin >> dividend;

    cout << "Enter divisor: ";

    cin >> divisor;

    quotient = dividend / divisor;

    remainder = dividend % divisor;

    cout << "Quotient = " << quotient << endl;

    cout << "Remainder = " << remainder;
```

```
    return 0;  
}
```

**OUTPUT:**

Enter dividend: 13

Enter divisor: 4

Quotient: 3

Remainder: 1

---

**10. Write a C++ program to manipulate the class account using classes and function. A user should be able to perform the following functions.**

---

**Deposit money.**

**Withdraw money.**

**Calculate the interest.**

**Check the total balance in his account.**

**AIM:** To manipulate the class account using classes and function.

**ALGORITHM:**

1. Start the program.
2. Declare the Class.
3. Define the variables.
4. Define the parameterized constructor.
5. Declare the function to deposit the amount. Enter the amount to be deposited.
6. Function to withdraw the amount.
7. Input the amount to be withdrawn. If the condition fails amount cannot be withdrawn.
8. Display the entire account details.
9. Stop the program.

**PROGRAM:**

```
#include<iostream.h>

#include<stdio.h>

#include<string.h>

using namespace std;

class bank

{
```

```

    int acno;

    char nm[100], acctype[100];

    float bal;

float p, t, r, interest;

    public:

        bank(int acc_no, char *name, char *acc_type, float balance)
//Parameterized Constructor

        {

            acno=acc_no;

            strcpy(nm, name);

            strcpy(acctype, acc_type);

            bal=balance;}

    void deposit();

    void withdraw();

    void display();

};

void bank::deposit() //depositing an amount

{

    int damt1;

    cout<<"\n Enter Deposit Amount = ";

    cin>>damt1;

    bal+=damt1;

}

void bank::withdraw() //withdrawing an amount

{

    int wamt1;

```

```

        cout<<"\n Enter Withdraw Amount = ";
        cin>>wamt1;
        if(wamt1>bal)
            cout<<"\n Cannot Withdraw Amount";
        bal-=wamt1;
    }
    void bank::display() //displaying the details
    {
        cout<<"\n -----";
        cout<<"\n Accout No. : "<<acno;
        cout<<"\n Name : "<<nm;
        cout<<"\n Account Type : "<<acctype;
        cout<<"\n Balance : "<<bal;
    }
    int main()
    {
        int acc_no;
        char name[100], acc_type[100];
        float balance;
        cout<<"\n Enter Details: \n";
        cout<<"-----";
        cout<<"\n Accout No. ";
        cin>>acc_no;
        cout<<"\n Name : ";
        cin>>name;

```

```

        cout<<"\n Account Type : ";

        cin>>acc_type;

        cout<<"\n Balance : ";

        cin>>balance;

        cout << "Enter principal amount, time and rate:";

        cin >> p >> t >> r;

        interest = (p*t*r)/100;

        cout << "Interest = " << interest << endl;

        bank b1(acc_no, name, acc_type, balance); //object is created

        b1.deposit(); //

        b1.withdraw(); // calling member functions

        b1.display(); //

        return 0;

}

```

### **OUTPUT:**

Enter Details:

-----

Account No. 45769812

Name: shiva

Account Type: Saving

Balance: 50000

Enter Deposit Amount: 5000

Enter Withdraw Amount: 10000

Enter principal amount, time and rate: 1000 10.5 11.9

-----



Account No. : 45769812

Name: shiva

Account Type: Saving

Balance: 45000

Interest: 1249.5

---

**11. Write a C++ program to generate prime numbers between 1 and 50.**

---

**Aim:** To generate prime numbers between 1 and 50.

**Algorithm:**

1. Create a loop to count numbers between 1 and 50.
2. Check each number if it is in the range.
3. Maintain factor count and increase factor count when found
4. Check if Prime or not. If prime print it.
5. Stop Execution.

**Program:**

```
#include <iostream.h>
#include <cmath.h>
using namespace std;
int main()
{ cout << "Prime Numbers between 1 and 50 are:\n";
  for(int i=2;i<=50;++i) //loop to check for each number in the range
  { int ctr=0; //to maintain factor count
    for(int j=2;j<=sqrt(i);++j) //checking for factors
    { if(i%j==0)
      ctr=1; //increasing factor count when found
    }
    if(ctr==0) //checking and printing prime numbers
      cout<<i<<" ";
  }
  return 0;
}
```

**Output:**

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

---

## 12. Write a C++ program to perform matrix addition and multiplication.

---

**Aim:** To perform matrix addition and multiplication.

**Algorithm:**

1. Initialize the required values and the size of the matrix.
2. Open for loops and Input the values for the matrix A
3. Close the For loops
4. Open For loops and Input the values for matrix B.
5. Close the For loops
6. Open For loops and add the matrix A and matrix B and store it in Matrix C.
7. Close the For loops
8. Open For loops and multiply the matrix A and matrix B and store it in Matrix C.
9. Close the For loops
10. Open For loops and print matrix C.
- 11 Close the For loops
12. Stop execution.

**PROGRAM:**

```
#include <iostream.h>

using namespace std;

int main()
{
//fill your code

int m, n;

cin >> m >> n;

int i, j;

int mat1[m][n], mat2[m][n], mat3[m][n];
```

```

for(i = 0; i < m; i++)
{
    for(j = 0; j < n; j++)
        cin >> mat1[i][j];
}

for(i = 0; i < n; i++)
{
    for(j = 0; j < n; j++)
    {
        cin >> mat2[i][j];
    }
}

for(i = 0; i < m; i++)
{
    for(j = 0; j < n; j++)
    {
        mat3[i][j] = mat1[i][j] + mat2[i][j];
    }
}

for(i = 0; i < m; i++)
{
    for(j = 0; j < n; j++)
        cout << mat3[i][j] << " ";
    cout << endl;
}

```

```
return 0;
```

```
}
```

**OUTPUT:**

2 2 (order of the matrix)

1 2 3 4 (matrix 1 elements)

2 3 4 5 (matrix 2 elements)

3 5 (resultant matrix)

7 9

---

**13. Write a C++ program to check whether the given matrix is a sparse matrix or not.**

---

**Aim:** To check whether the given matrix is a sparse matrix or not.

**Algorithm:**

1. Declare and initialize a two-dimensional array a.
2. Calculate the number of rows and columns present in the given array and store it in variables rows and cols respectively.
3. Loop through the array and count the number of zeroes present in the given array and store in the variable count.
4. Calculate the size of the array by multiplying the number of rows with many columns of the array.
5. If the count is greater than size/2, given matrix is the sparse matrix. That means, most of the elements of the array are zeroes.
6. Else, the matrix is not a sparse matrix.

**Program:**

```
#include<iostream.h>

using namespace std;

int main () {

    int a[10][10] = { {2, 0, 0} , {0, 3, 8} , {0, 9, 0} };

    int i, j, count = 0;

    int r = 3, c = 3;

    for (i = 0; i < r; ++i) {

        for (j = 0; j < c; ++j) {

            if (a[i][j] == 0)

                count++;

        }

    }

}
```

```

cout<<"The matrix is:"<<endl;
for (i = 0; i < r; ++i) {
    for (j = 0; j < c; ++j) {
        cout<<a[i][j]<<" ";
    }
    cout<<endl;
}

cout<<"There are "<<count<<" zeros in the matrix"<<endl;
if (count > ((r * c)/ 2))
    cout<<"This is a sparse matrix"<<endl;
else
    cout<<"This is not a sparse matrix"<<endl;
return 0;
}

```

**Output:**

The matrix is:

2 0 0

0 3 8

0 9 0

There are 5 zeros in the matrix

This is a sparse matrix

---

#### 14. Write a C++ program to overload unary minus operator.

---

**AIM:** To overload unary minus operator.

**ALGORITHM:**

- 1: Start the program.
- 2: Declare the class.
- 3: Declare the variables and its member function.
4. Define the Constructor.
5. Display the Distance.
6. Call the function operator - -() by decrementing the class object and call the function display.
7. Stop the program.

**Program:**

```
#include <iostream.h>

using namespace std;

class Distance {

    private:

        int feet;

        int inches;

    public:

        // Constructor

        Distance(int f, int i) {

            feet = f;

            inches = i;

        }

        // method to display distance

        void display() {
```



```

        cout << "F: " << feet << " I:" << inches << endl;
    }

    // overloaded minus(-) operator
    Distance operator-() {
        feet = -feet;
        inches = -inches;
        return Distance(feet, inches);
    }
};

int main() {
    Distance D1(3, 4), D2(-1, 10);

    !D1;
    D1.display();

    !D2;
    D2.display();

    return 0;
}

```

**Output:**

This will give the output –

F: -3 I:-4

F: 1 I:-10

---

**15. Write a C++ program to calculate total sales and average sales made by a salesman.**

---

**AIM: To calculate total sales and average sales made by a salesman.**

Here, we are to calculate the total sales amount and the sales quantity made by a salesman.

**ALGORITHM:**

1. Start the program.
2. Define and declare the variables.
3. Check if the sales price is higher than the quantity.
4. Execute the condition.
5. Display the total sales and average sales by a salesman.
6. If the condition is not satisfied return 0.
7. Stop the program.

**Program:**

```
#include <iostream.h>

using namespace std;

int main()
{
    int sale_price, qty, ctr = 0, sum1 = 0, sum2 = 0;
    cout << "Input Sales Price and Sales Quantity: ";
    while (cin >> sale_price >> qty)
    {
        sum1 += sale_price * qty;
        sum2 += qty;
        ctr++;
    }
}
```

```
    }  
  
    cout << "\nTotal of the sales amount and the average of the sales  
quantity:\n";  
  
    cout << sum1 << endl << static_cast<int>(static_cast<double>(sum2) /  
ctr + 0.5) << endl;  
  
    return 0;  
  
}
```

**Output:**

Input: 2 5

Input Sales Price and Sales Quantity:

Total of the sales amount and the average of the sales quantity:













10

5

## Document Information

Analyzed document	c lab manual 2.6.2022 Final.docx (D139108403)
Submitted	2022-06-03T08:25:00.0000000
Submitted by	kalaiaarasi
Submitter email	kalaiaarasidr@gmail.com
Similarity	25%
Analysis address	kalaiaarasidr.tnou@analysis.urkund.com

## Sources included in the report

W	URL: <a href="https://learn.saylor.org/mod/book/view.php?id=33554&amp;chapterid=18263">https://learn.saylor.org/mod/book/view.php?id=33554&amp;chapterid=18263</a> Fetched: 2022-05-22T07:15:26.5770000	 2
W	URL: <a href="http://docplayer.info/226839625-Modul-praktik-ta-2020-2021.html">http://docplayer.info/226839625-Modul-praktik-ta-2020-2021.html</a> Fetched: 2022-06-03T08:25:36.7430000	 4
W	URL: <a href="https://iqcode.com/code/cpp/fibonacci-series-in-c">https://iqcode.com/code/cpp/fibonacci-series-in-c</a> Fetched: 2022-06-03T08:25:57.1500000	 1
W	URL: <a href="https://foroayuda.es/escriba-un-programa-para-generar-series-de-fibonacci-en-un-rango-dado-ejemplo-de-codigo-c/">https://foroayuda.es/escriba-un-programa-para-generar-series-de-fibonacci-en-un-rango-dado-ejemplo-de-codigo-c/</a> Fetched: 2022-06-03T08:25:40.0530000	 1
W	URL: <a href="https://www.programiz.com/cpp-programming/examples/fibonacci-series">https://www.programiz.com/cpp-programming/examples/fibonacci-series</a> Fetched: 2020-09-28T10:22:25.0600000	 1
SA	<b>15691A0324.docx</b> Document 15691A0324.docx (D19320821)	 1
W	URL: <a href="http://www.drebrahimi.com/chapter17.html">http://www.drebrahimi.com/chapter17.html</a> Fetched: 2021-02-19T16:54:47.0500000	 1
W	URL: <a href="https://rumenz.com/examples/cpp/cplusplus-concatenate-two-strings.html">https://rumenz.com/examples/cpp/cplusplus-concatenate-two-strings.html</a> Fetched: 2022-06-03T08:25:32.1600000	 2
W	URL: <a href="https://coderforevers.com/cpp/cpp-program/quotient-remainder/">https://coderforevers.com/cpp/cpp-program/quotient-remainder/</a> Fetched: 2021-02-28T19:56:05.4700000	 2
SA	<b>CPR202_12180061.docx</b> Document CPR202_12180061.docx (D84199828)	 2
W	URL: <a href="https://programiranjec.weebly.com/zadaci">https://programiranjec.weebly.com/zadaci</a> Fetched: 2021-12-11T00:31:11.1700000	 1
W	URL: <a href="https://www.tutorialspoint.com/cplusplus-program-to-perform-matrix-multiplication">https://www.tutorialspoint.com/cplusplus-program-to-perform-matrix-multiplication</a> Fetched: 2020-04-12T23:32:20.7730000	 1



## About Tamil Nadu Open University



Tamil Nadu Open University was established in 2002 by an Act of Tamil Nadu Legislature, with the objective of introducing and promoting Open University and Distance Education systems in Tamil Nadu. Relaxed entry procedures, maintenance of standards, Individualized study, flexibility in terms of place, duration of study, use of latest ICT, well-knit student support services network, cost effective programmes, collaboration and resource sharing with other Universities are its salient features. Presently functioning at its headquarters at Saidapet, Chennai.

### School of Computer Science

The School of Computer Science (SOCS) was established on 24-10-2005 with the objective of imparting quality education in the field of Computer Science. The broad vision of the School is to empower the students with the required skills to deliver quality, prompt, effective problem solver to the society. The School offers a wide array of research opportunities and programs of study at Under Graduate and Post Graduate level. The various programmes offered under the school of Computer Science are as follows:

#### Under Graduate Programmes

- Bachelor of Computer Applications (B.C.A.)
- Bachelor of Computer Applications-Lateral Entry (B.C.A.-LE)
- Bachelor of Science in Computer Science (B.Sc-CS)

#### Post Graduate Programmes

- Master of Computer Applications (M.C.A.)
- Master of Computer Applications-Lateral Entry (M.C.A-LE)
- Master of Science in Computer Science (M.Sc-CS)

#### Research Programmes (Computer Science)

- Ph.D., (Full Time/Part Time)
- M.Phil., (Full Time/Part Time)



# Tamil Nadu Open University

CHENNAI - 600 015

[www.tnou.ac.in](http://www.tnou.ac.in)