

EXERCISE – 01

Date- 20/01/2022

SIMPLE C++ PROGRAMS

Objective-

To understand simple C++ programs and to demonstrate various decision making and loop constructs.

Example1-

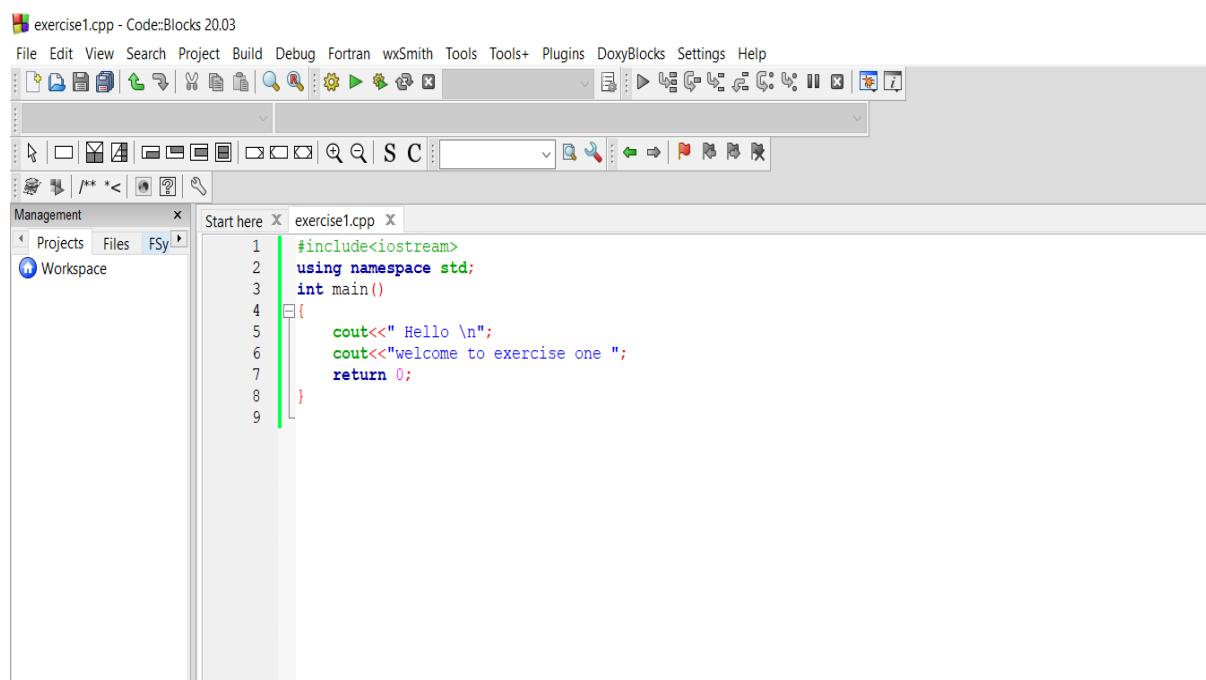
Simple program in C++ language for printing “hello” and “welcome to exercise one”.

Aim: To understand the use of ‘cout’ statement in simple program.

Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Use cout statement to print “Hello” and “Welcome to exercise one”
- Exit .

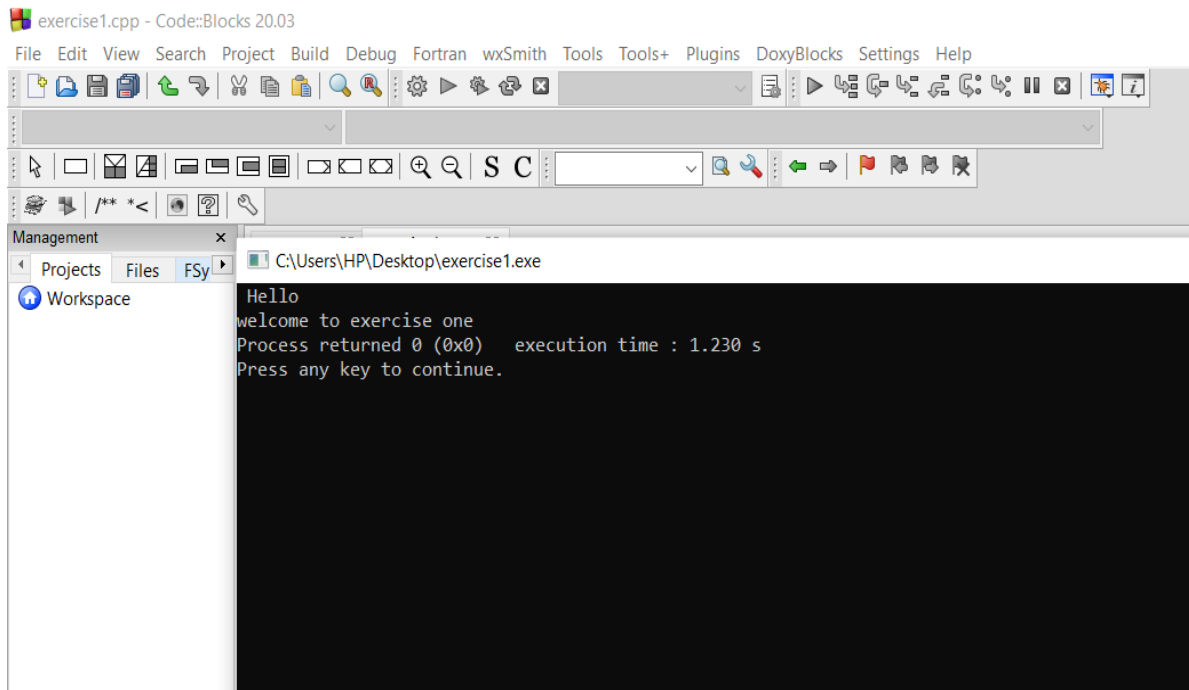
Input-



The screenshot shows a C++ IDE with the following code in exercise1.cpp:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     cout<<" Hello \n";
6     cout<<"welcome to exercise one ";
7     return 0;
8 }
9
```

Output-



Example 2-

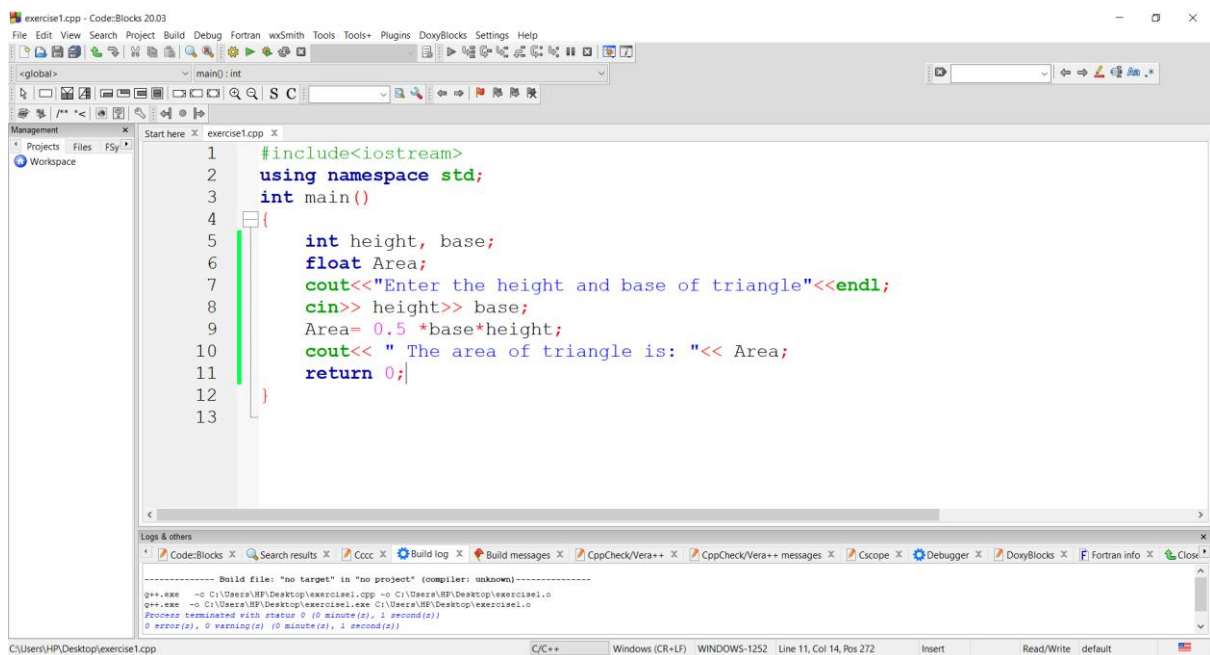
A general program to find the area of triangle with height “H” and base “B”.

Aim- To understand how to use the arithmetic operators in various programs.

Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Start
- Declare the type of data
- Use cout statement to print “Enter the height and base of triangle”
- Use cin to take values from users
- Use formula $\text{area} = 0.5 * \text{base} * \text{height}$
- Use cout statement to show the area calculated
- End

Input-

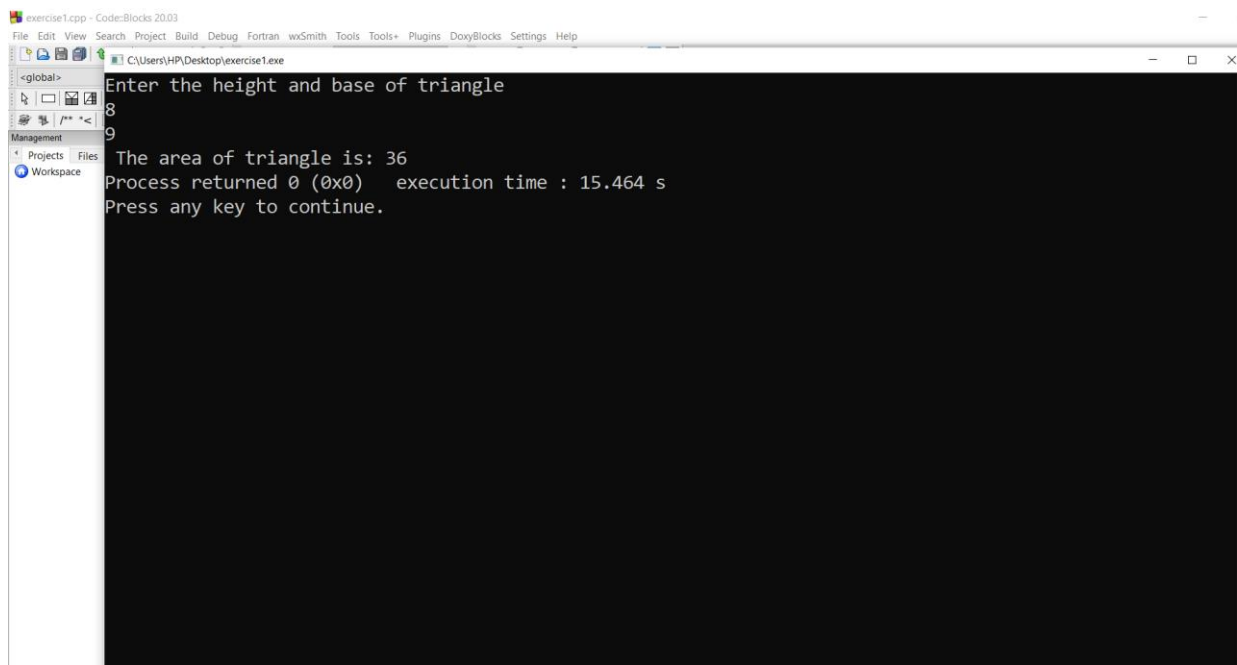


```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int height, base;
6      float Area;
7      cout<<"Enter the height and base of triangle"<<endl;
8      cin>> height>> base;
9      Area= 0.5 *base*height;
10     cout<< " The area of triangle is: "<< Area;
11     return 0;
12 }
13
```

Build file: "no target" in "no project" (compiler: unknown)

g++ -x86_64-linux-gnu -c C:\Users\HP\Desktop\exercise1.cpp -o C:\Users\HP\Desktop\exercise1.o
g++ -x86_64-linux-gnu -o C:\Users\HP\Desktop\exercise1.exe C:\Users\HP\Desktop\exercise1.o
Process terminated with status 0 (0 minute(s), 1 second(s))
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))

Output-



```
Enter the height and base of triangle
8
9
The area of triangle is: 36
Process returned 0 (0x0)   execution time : 15.464 s
Press any key to continue.
```

Example 3-

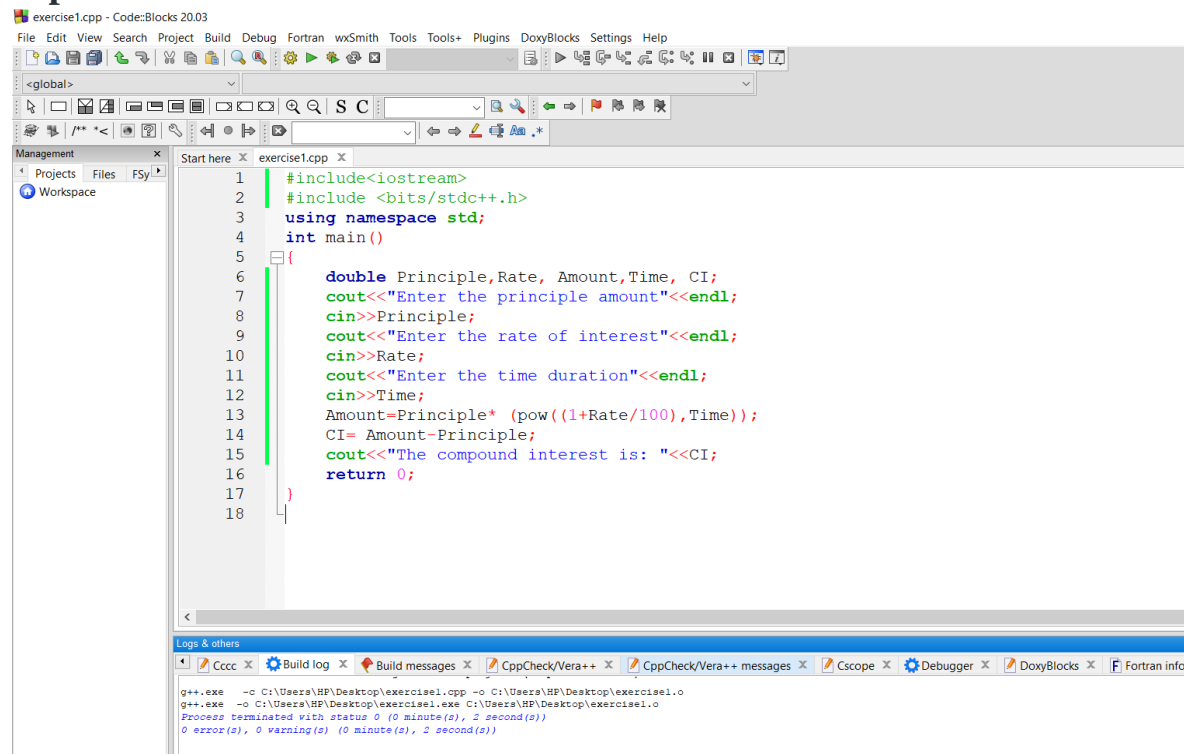
A general program to find the compound interest using different arithmetic operators

Aim- To understand how to use the arithmetic operators in various programs.

Algorithm-

- Enter the pre-processor directive “include <iostream>” and “include <bits/stdc++.h>”
- Write using namespace std;
- Enter the main()
- Start
- Input the principle amount in some variable-principle.
- Input time in some variable-time.
- Input rate in some variable -rate.
- Calculate Amount using formula,
- $\text{Amount} = \text{principle} * (1 + \text{rate} / 100)^{\text{time}}$.
- Calculate Compound Interest using Formula.
- Print the value of CI using cout statement
- Exit

Input-



The screenshot shows a C++ IDE with the following code in exercise1.cpp:

```
1 #include<iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4 int main()
5 {
6     double Principle,Rate, Amount,Time, CI;
7     cout<<"Enter the principle amount"<<endl;
8     cin>>Principle;
9     cout<<"Enter the rate of interest"<<endl;
10    cin>>Rate;
11    cout<<"Enter the time duration"<<endl;
12    cin>>Time;
13    Amount=Principle* (pow((1+Rate/100),Time));
14    CI= Amount-Principle;
15    cout<<"The compound interest is: "<<CI;
16    return 0;
17 }
18
```

The output window shows the following execution details:

```
g++.exe -o C:\Users\HP\Desktop\exercisel.exe -o C:\Users\HP\Desktop\exercisel.o
g++.exe -o C:\Users\HP\Desktop\exercisel.exe C:\Users\HP\Desktop\exercisel.o
Process terminated with status 0 (0 minute(s), 2 second(s))
0 error(s), 0 warning(s) (0 minute(s), 2 second(s))
```

Output-

C:\Users\HP\Desktop\exercise1.exe

```
Enter the principle amount
3000
Enter the rate of interest
8.5
Enter the time duration
4
The compound interest is: 1157.58
Process returned 0 (0x0)   execution time : 14.595 s
Press any key to continue.
```

Example 4-

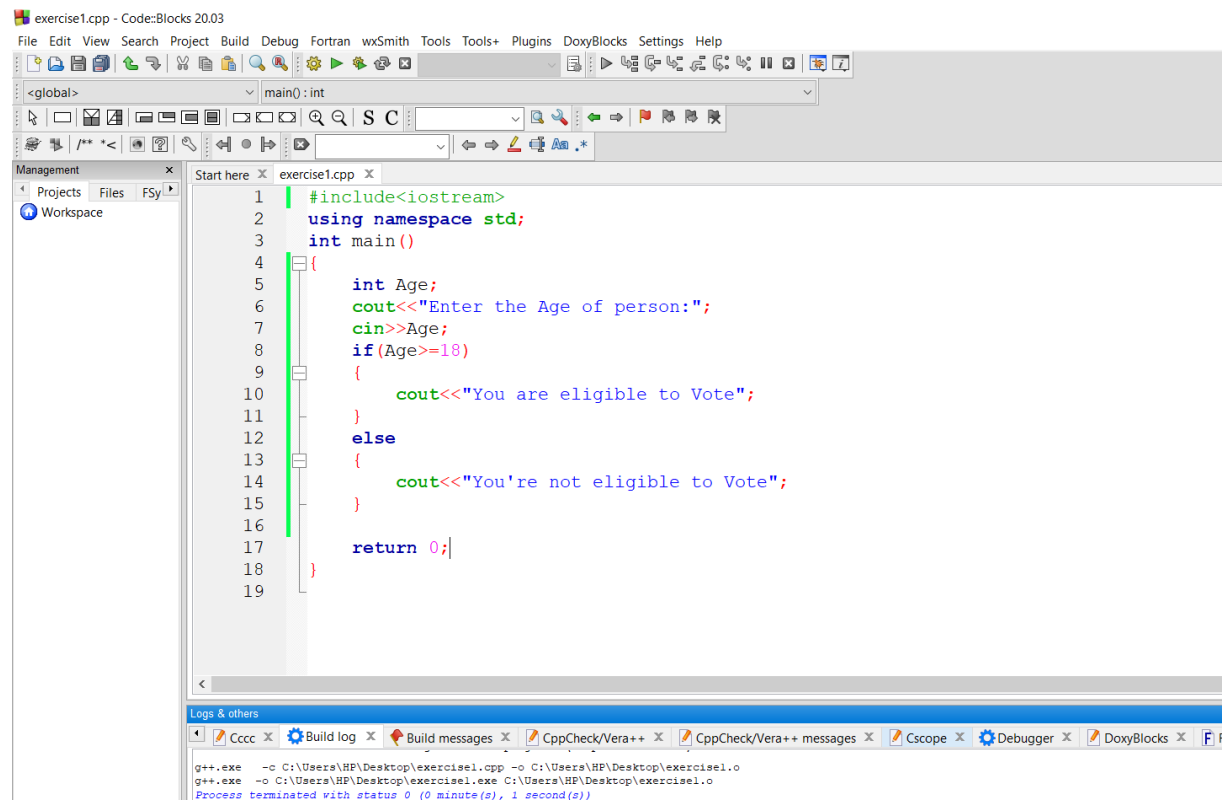
A program to find whether a person is eligible to vote or not by using simple 'if-else' statement.

AIM- to understand a simple if-else construct

Algorithm-

- Enter the pre-processor directive "include <iostream>"
- Write using namespace std;
- Enter the main()
- Start
- Declare the type of data
- Take the input of data using cin
- If age >= 18, print "you are eligible to vote"
- Else , "not eligible"
- Exit

Input-



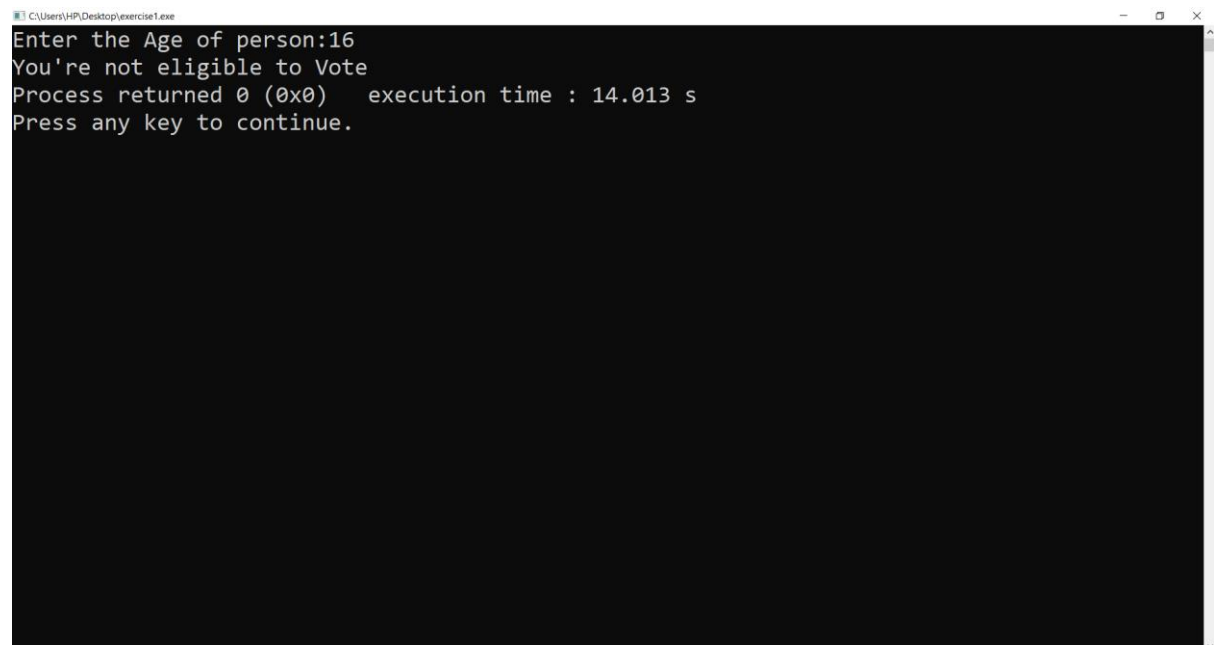
```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int Age;
6      cout<<"Enter the Age of person:";
7      cin>>Age;
8      if(Age>=18)
9      {
10         cout<<"You are eligible to Vote";
11     }
12     else
13     {
14         cout<<"You're not eligible to Vote";
15     }
16
17     return 0;
18 }
19
```

Logs & others

Cccc x Build log x Build messages x CppCheck/Vera++ x CppCheck/Vera++ messages x Cscope x Debugger x DoxyBlocks x

g++.exe -o C:\Users\HP\Desktop\exercisel.exe -o C:\Users\HP\Desktop\exercisel.o C:\Users\HP\Desktop\exercisel.cpp
g++.exe -o C:\Users\HP\Desktop\exercisel.exe C:\Users\HP\Desktop\exercisel.o
Process terminated with status 0 (0 minute(s), 1 second(s))

Output-



```
C:\Users\HP\Desktop\exercisel.exe
Enter the Age of person:16
You're not eligible to Vote
Process returned 0 (0x0)   execution time : 14.013 s
Press any key to continue.
```

```
C:\Users\HP\Desktop\exercise1.exe
Enter the Age of person:32
You are eligible to Vote
Process returned 0 (0x0)   execution time : 10.125 s
Press any key to continue.
```

Example 5-

A program to find nature of amino acid by using nested 'if-else' statement.

AIM- to understand a nested if-else construct.

Algorithm-

- Enter the pre-processor directive "include <iostream>"
- Write using namespace std;
- Enter the main()
- Start
- Declare the type of data
- If marks>=85, print "Grade is A"
- elseif marks>=70, print "Grade is B"
- elseif marks>=55, print "Grade is C"
- elseif marks>=40, print "Grade is D"
- elseif marks>=30, print "Grade is E"
- else, print "FAIL"
- Exit

Input-

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int m;
6      cout<<"Enter percentage of marks obtained:";
7      cin>>m;
8      if(m>85)
9      {
10         cout<<"Grade is A";
11     }
12     else if(m>70)
13     {
14         cout<<"Grade is B";
15     }
16     else if(m>55)
17     {
18         cout<<"Grade is C";
19     }
20     else if(m>40)
21     {
22         cout<<"Grade is D";
23     }
24     else if(m>30)
25     {
26         cout<<"Grade is E";
27     }
28     else{
29         cout<<"FAIL";
30     }
31 }
```

Output-

```
C:\Users\HP\Desktop\exercise1.exe
Enter percentage of marks obtained:76
Grade is B
Process returned 0 (0x0)   execution time : 7.673 s
Press any key to continue.
```



```
C:\Users\HP\Desktop\exercice1.exe
Enter percentage of marks obtained:37
Grade is E
Process returned 0 (0x0)   execution time : 6.104 s
Press any key to continue.
```

```
C:\Users\HP\Desktop\exercice1.exe
Enter percentage of marks obtained:97
Grade is A
Process returned 0 (0x0)   execution time : 8.061 s
Press any key to continue.
```

EXERCISE – 02

Date- 03/02/2022

Working with matrices

Objective-

To understand C++ programs dealing with matrices.

Example1-

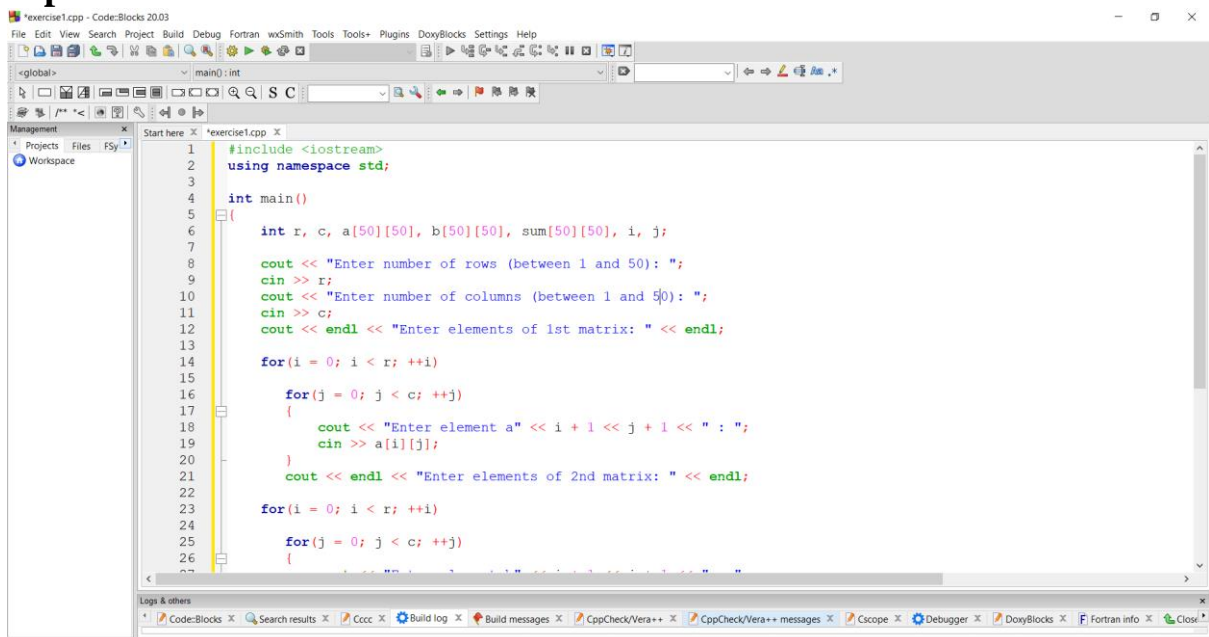
A general program to find the sum of two matrices.

Aim: To understand how to work with matrices

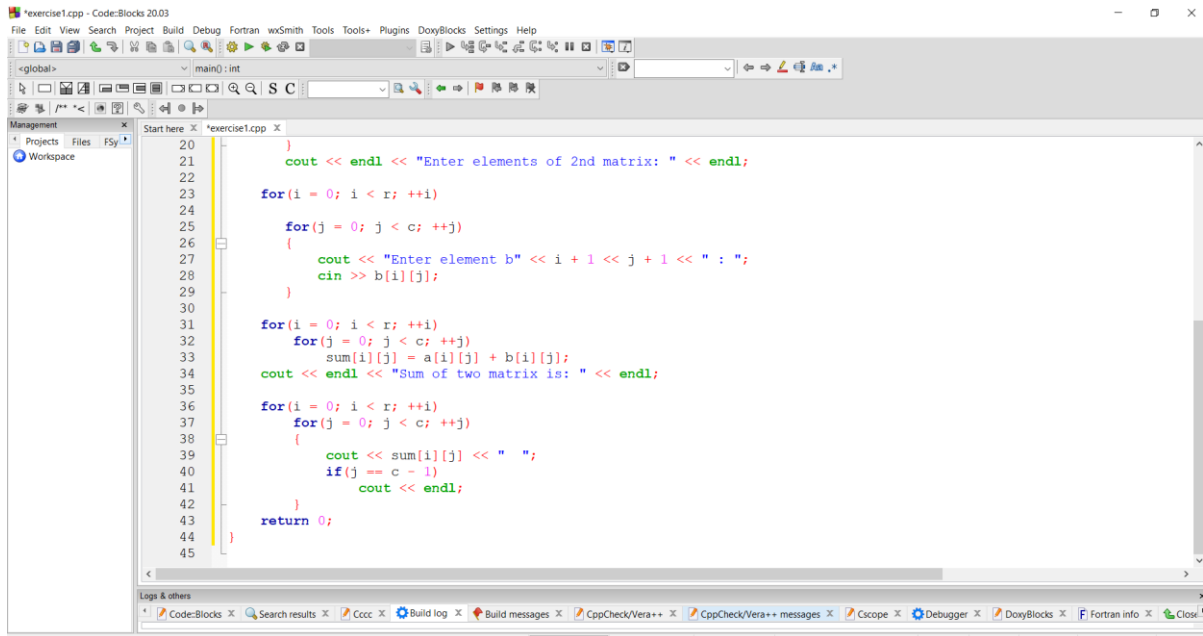
Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Declare the array, transpose array
- Storing elements of first matrix
- Storing elements of second matrix
- Adding Two matrices
- Displaying the resultant sum matrix.
- Exit .

Input-



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int r, c, a[50][50], b[50][50], sum[50][50], i, j;
7
8     cout << "Enter number of rows (between 1 and 50): ";
9     cin >> r;
10    cout << "Enter number of columns (between 1 and 50): ";
11    cin >> c;
12    cout << endl << "Enter elements of 1st matrix: " << endl;
13
14    for(i = 0; i < r; ++i)
15    {
16        for(j = 0; j < c; ++j)
17        {
18            cout << "Enter element a" << i + 1 << j + 1 << " : ";
19            cin >> a[i][j];
20        }
21        cout << endl << "Enter elements of 2nd matrix: " << endl;
22    }
23
24    for(i = 0; i < r; ++i)
25    {
26        for(j = 0; j < c; ++j)
```



Output-

```

C:\Users\HP\Desktop\exercise1.exe
Enter number of rows (between 1 and 50): 4
Enter number of columns (between 1 and 50): 3

Enter elements of 1st matrix:
Enter element a11 : 1
Enter element a12 : 2
Enter element a13 : 3
Enter element a21 : 4
Enter element a22 : 5
Enter element a23 : 6
Enter element a31 : 7
Enter element a32 : 8
Enter element a33 : 9
Enter element a41 : 10
Enter element a42 : 11
Enter element a43 : 12

Enter elements of 2nd matrix:
Enter element b11 : 11
Enter element b12 : 13
Enter element b13 : 12
Enter element b21 : 14
Enter element b22 : 15
Enter element b23 : 16
Enter element b31 : 17
Enter element b32 : 18
Enter element b33 : 19
Enter element b41 : 20
Enter element b42 : 10
Enter element b43 : 20

Sum of two matrix is:
12 15 15
18 20 22
24 26 28
30 21 32

Process returned 0 (0x0)   execution time : 89.007 s
Press any key to continue.

```

Example2-

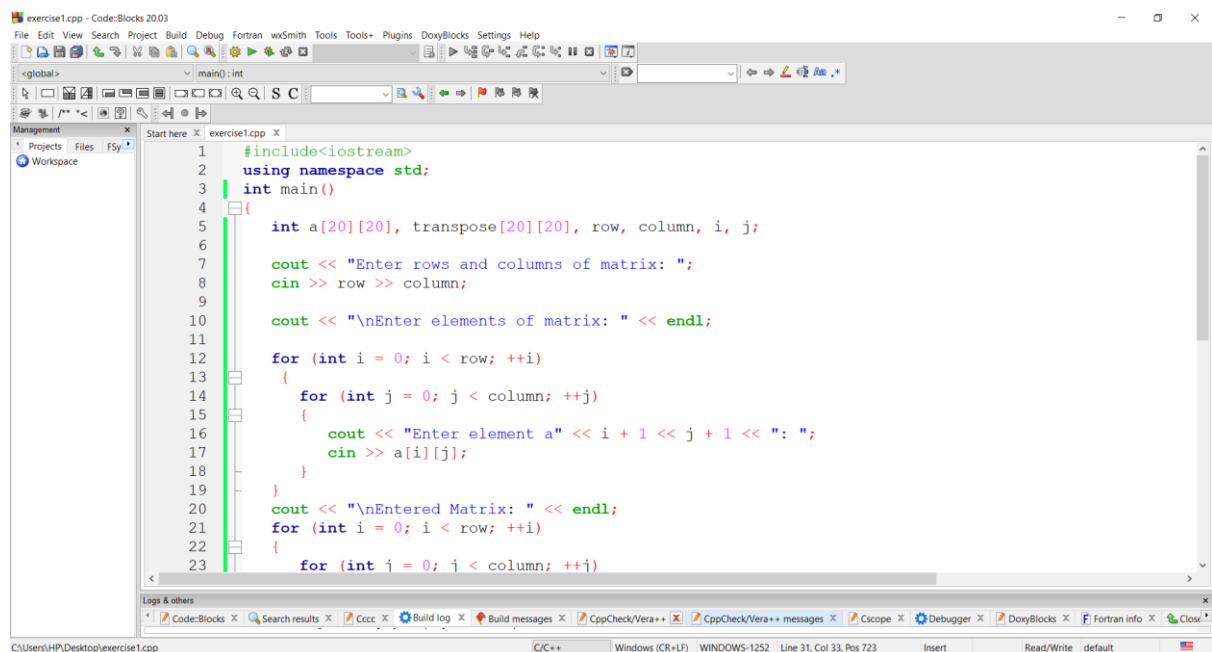
A general program to find the transpose of a matrix.

Aim: To understand how to work with matrices

Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Declare the array, transpose array
- Storing matrix elements
- Printing the a matrix
- Computing transpose of the matrix
- Printing the transpose
- Exit .

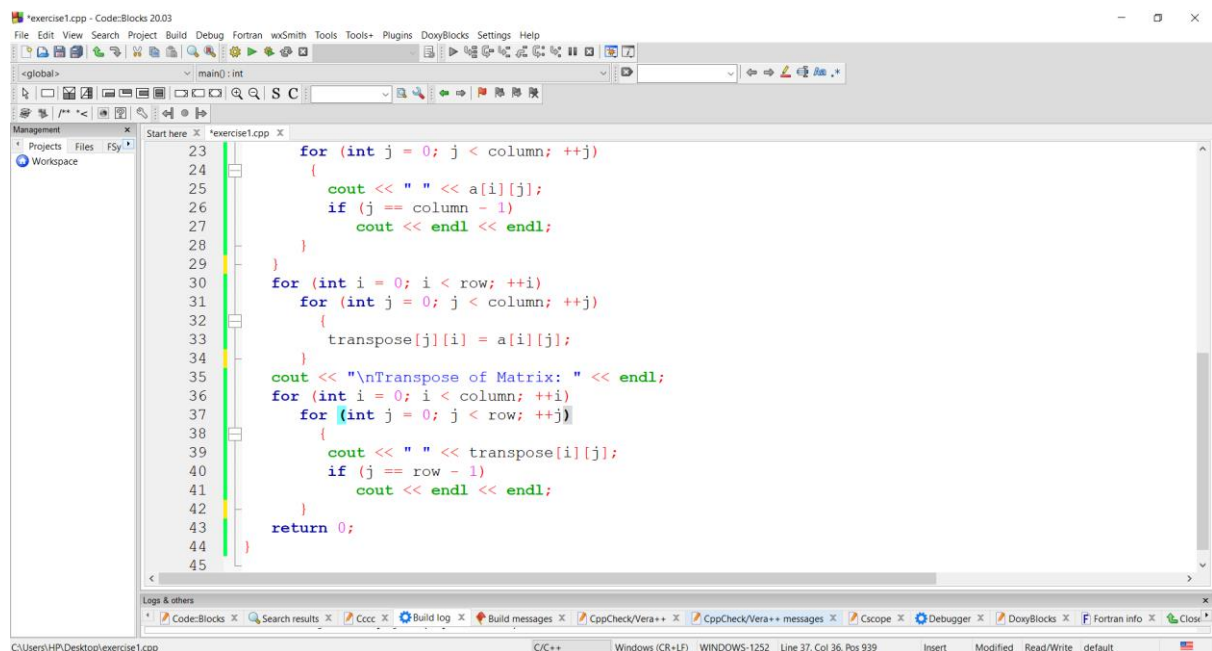
Input-



```

1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a[20][20], transpose[20][20], row, column, i, j;
6
7     cout << "Enter rows and columns of matrix: ";
8     cin >> row >> column;
9
10    cout << "\nEnter elements of matrix: " << endl;
11
12    for (int i = 0; i < row; ++i)
13    {
14        for (int j = 0; j < column; ++j)
15        {
16            cout << "Enter element a" << i + 1 << j + 1 << ": ";
17            cin >> a[i][j];
18        }
19    }
20    cout << "\nEntered Matrix: " << endl;
21    for (int i = 0; i < row; ++i)
22    {
23        for (int j = 0; j < column; ++j)

```



```

23        for (int j = 0; j < column; ++j)
24        {
25            cout << " " << a[i][j];
26            if (j == column - 1)
27                cout << endl << endl;
28        }
29    }
30    for (int i = 0; i < row; ++i)
31    {
32        for (int j = 0; j < column; ++j)
33        {
34            transpose[j][i] = a[i][j];
35        }
36    }
37    cout << "\nTranspose of Matrix: " << endl;
38    for (int i = 0; i < column; ++i)
39    {
40        for (int j = 0; j < row; ++j)
41        {
42            cout << " " << transpose[i][j];
43            if (j == row - 1)
44                cout << endl << endl;
45        }
46    }
47    return 0;
48 }

```

Output-

```
C:\Users\HP\Desktop\exercise1.exe
Enter rows and columns of matrix: 2
5

Enter elements of matrix:
Enter element a11: 2
Enter element a12: 4
Enter element a13: 7
Enter element a14: 9
Enter element a15: 2
Enter element a21: 5
Enter element a22: 18
Enter element a23: 45
Enter element a24: 32
Enter element a25: 65

Entered Matrix:
2 4 7 9 2

5 18 45 32 65

Transpose of Matrix:
2 5

4 18

7 45

9 32

2 65

Process returned 0 (0x0)   execution time : 50.990 s
Press any key to continue.
```

EXERCISE – 03

Date- 10/02/2022

Demonstration of switch construct

Objective-

To demonstrate some programs of switch construct.

Example1-

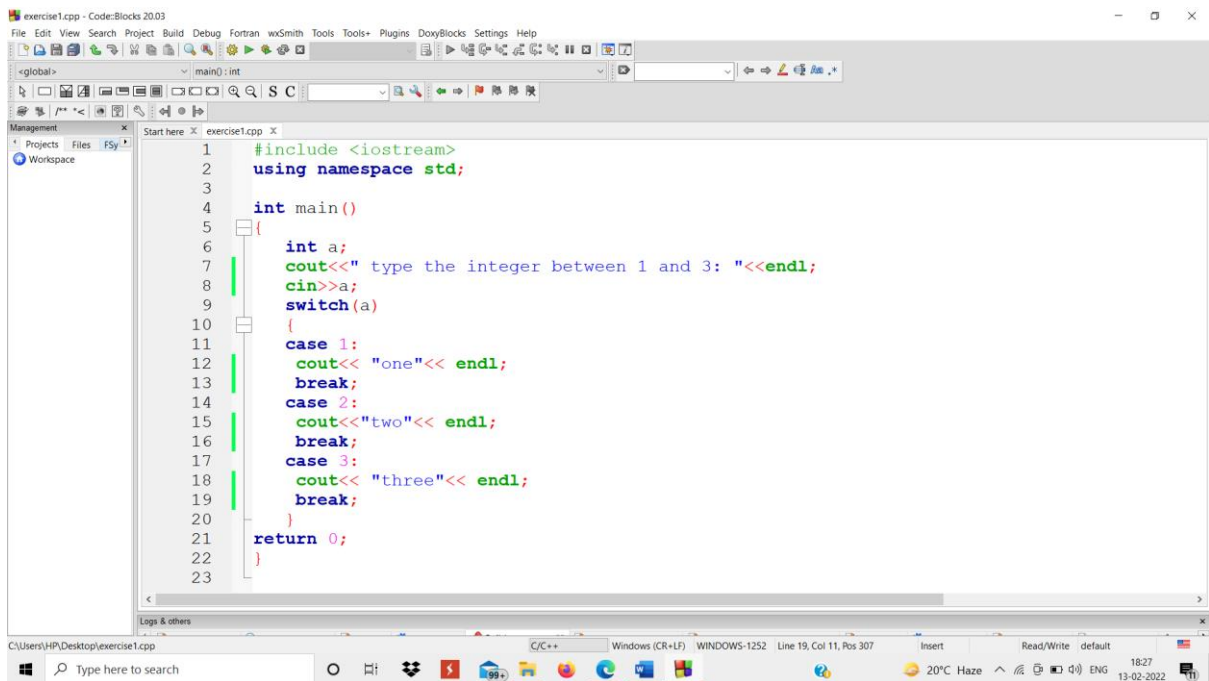
A general program of switch construct.

Aim: To understand implementation of switch construct

Algorithm-

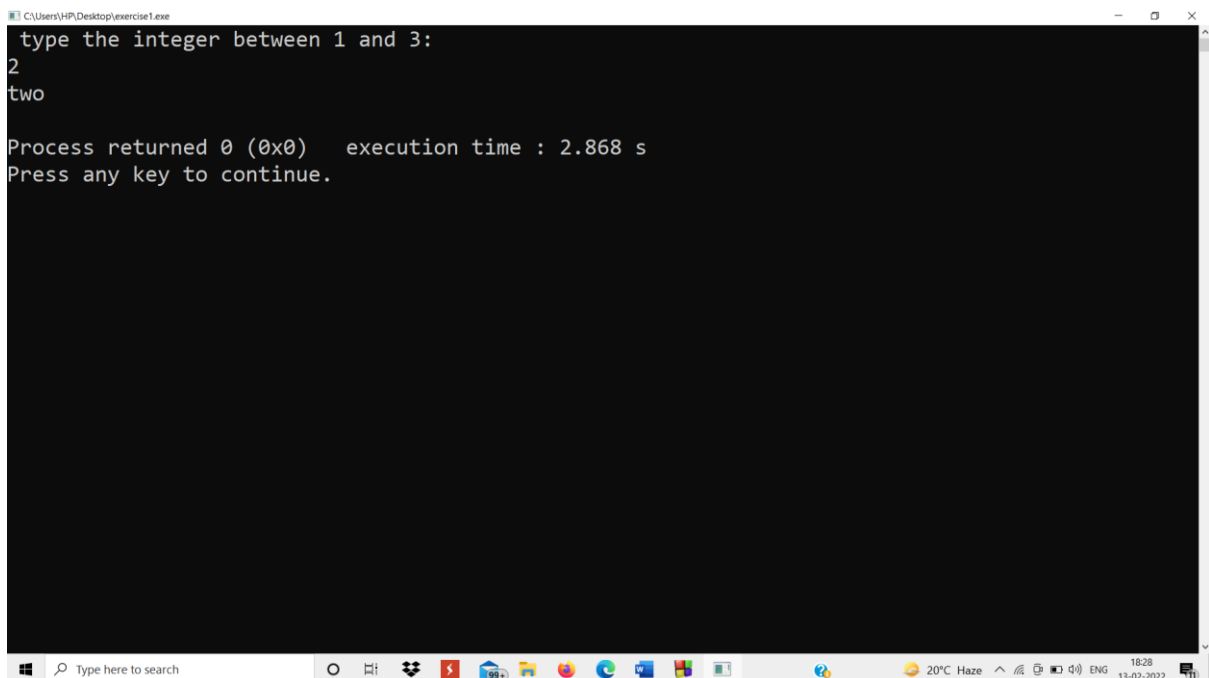
- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Declare a variable “a” value between 1 and 3
- Take the value of a from user
- Case 1: cout<< “one”<< endl;
- Break;
- Case 2: cout<< “two”<< endl;
- Break;
- Case 3: cout<< “three”<< endl;
- Break;
- Exit .

Input-



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a;
7     cout<<" type the integer between 1 and 3: "<<endl;
8     cin>>a;
9     switch(a)
10    {
11        case 1:
12            cout<<"one"<< endl;
13            break;
14        case 2:
15            cout<<"two"<< endl;
16            break;
17        case 3:
18            cout<<"three"<< endl;
19            break;
20    }
21    return 0;
22 }
23
```

Output-



```
type the integer between 1 and 3:
2
two

Process returned 0 (0x0)   execution time : 2.868 s
Press any key to continue.
```

Example2-

A general program for making a calculator using switch construct.

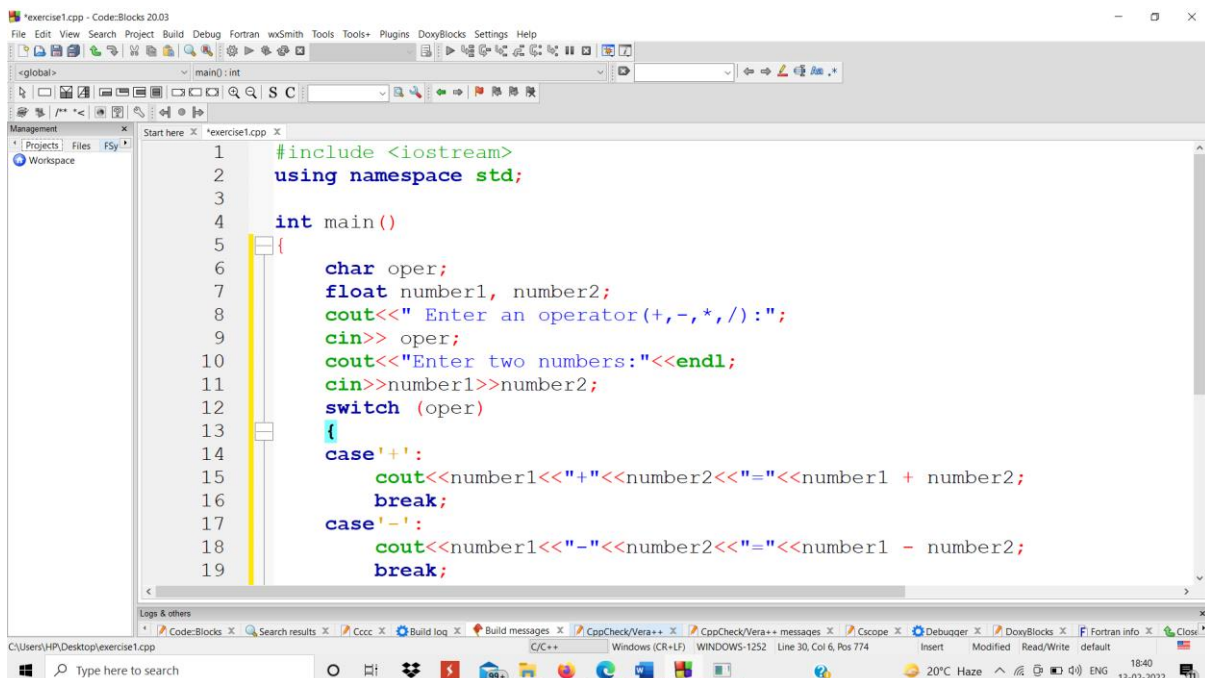
Aim: To make a general calculator using switch construct

Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;

- Enter the main()
- Declare the type of operator and number
- Take values from user for number and operator
- switch (oper)
- {
- case '+':
- cout<<number1<<"+"<<number2<<"="<<number1 + number2;
- break;
- case '-':
- cout<<number1<<"-"<<number2<<"="<<number1 - number2;
- break;
- case '*':
- cout<<number1<<"*"<<number2<<"="<<number1 * number2;
- break;
- case '/':
- cout<<number1<<"/"<<number2<<"="<<number1 / number2;
- break;
- default:
- //operator doesn't match any case constant (+,-,*,/)
- cout<<"Error! The operator is incorrect";
- break;
- }Exit .

Input-



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char oper;
7      float number1, number2;
8      cout<<" Enter an operator(+, -, *, /) : ";
9      cin>> oper;
10     cout<<"Enter two numbers:"<<endl;
11     cin>>number1>>number2;
12     switch (oper)
13     {
14     case '+':
15         cout<<number1<<"+"<<number2<<"="<<number1 + number2;
16         break;
17     case '-':
18         cout<<number1<<"-"<<number2<<"="<<number1 - number2;
19         break;

```



```
15      cout<<number1<<"+"<<number2<<"="<<number1 + number2;
16      break;
17      case '-':
18      cout<<number1<<"-"<<number2<<"="<<number1 - number2;
19      break;
20      case '*':
21      cout<<number1<<"*"<<number2<<"="<<number1 * number2;
22      break;
23      case '/':
24      cout<<number1<<"/"<<number2<<"="<<number1 / number2;
25      break;
26      default:
27      //operator doesn't match any case constant (+,-,*,/)
28      cout<<"Error! The operator is incorrect";
29      break;
30  }
31  return 0;
32  }
33  }
```

Output-

```
Enter an operator(+,-,*,/):*
Enter two numbers:
4
9
4*9=36
Process returned 0 (0x0)   execution time : 34.229 s
Press any key to continue.
```

EXERCISE – 04

Date- 10/02/2022

USER DEFINED FUNCTIONS

Objective-

To understand working of user defined functions.

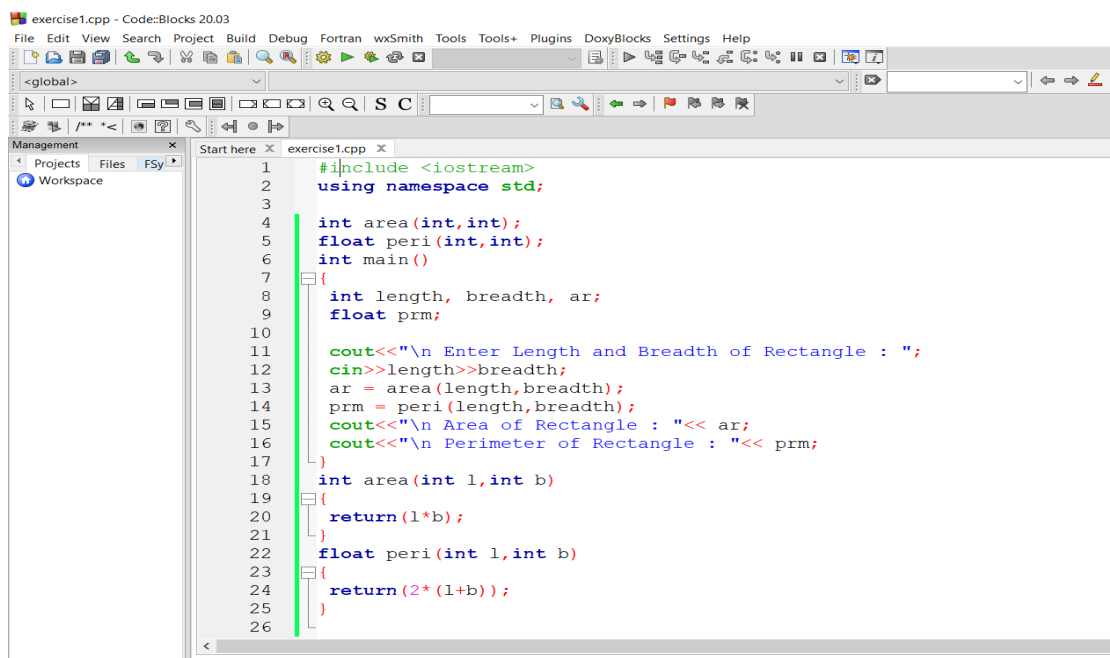
Example1- a program to calculate area and perimeter of square and rectangle. Two functions for area and perimeter respectively and accept the required measurements from the user

Aim: To generate a program to calculate area and perimeter of square and rectangle.

Algorithm-

- Enter the pre-processor directive “include <iostream>”
- Write using namespace std;
- Enter the main()
- Accept the length of rectangle
- Accept the breadth of rectangle
- Compute the area of rectangle with help of formula $A = \text{length} * \text{breadth}$
- Compute the perimeter with help of formula $P = 2 * (L + B)$
- Display result
- Stop.

Input-



```
exercise1.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Workspace
Start here exercise1.cpp
1  #include <iostream>
2  using namespace std;
3
4  int area(int,int);
5  float peri(int,int);
6  int main()
7  {
8      int length, breadth, ar;
9      float prm;
10
11     cout<<"\n Enter Length and Breadth of Rectangle : ";
12     cin>>length>>breadth;
13     ar = area(length,breadth);
14     prm = peri(length,breadth);
15     cout<<"\n Area of Rectangle : "<< ar;
16     cout<<"\n Perimeter of Rectangle : "<< prm;
17 }
18 int area(int l,int b)
19 {
20     return (l*b);
21 }
22 float peri(int l,int b)
23 {
24     return (2*(l+b));
25 }
26
```

Output-

C:\Users\HP\Desktop\exercise1.exe

```
Enter Length and Breadth of Rectangle : 34
76

Area of Rectangle : 2584
Perimeter of Rectangle : 220
Process returned 0 (0x0)   execution time : 18.976 s
Press any key to continue.
```

EXERCISE – 05

Date- 10/02/2022

Working with pointers

Objective-

To demonstrate some programs of working with pointers.

Example-

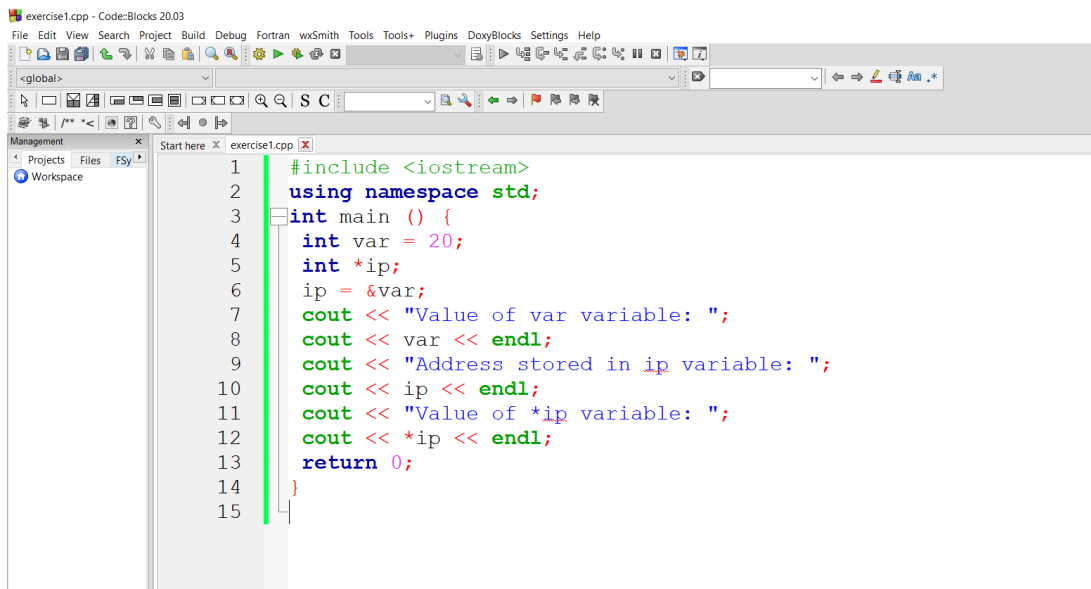
A general program of C++ to demonstrate the concept of pointers, including assigning, storing and accessing values of pointers.

Aim: To understand implementation of pointers.

Algorithm-

1. Enter the pre-processor directive “include <iostream>”
2. Write using namespace std;
3. Enter the main()
4. actual variable declaration
5. pointer variable
6. store address of var in pointer variable
7. print value of variable
8. print the address stored in ip pointer variable
9. access the value at the address available in pointer
- 10.Exit

Input-



```
1  #include <iostream>
2  using namespace std;
3  int main () {
4      int var = 20;
5      int *ip;
6      ip = &var;
7      cout << "Value of var variable: ";
8      cout << var << endl;
9      cout << "Address stored in ip variable: ";
10     cout << ip << endl;
11     cout << "Value of *ip variable: ";
12     cout << *ip << endl;
13     return 0;
14 }
15
```

Output-

C:\Users\HP\Desktop\exercise1.exe

Value of var variable: 20

Address stored in ip variable: 0x61fe14

Value of *ip variable: 20

Process returned 0 (0x0) execution time : 0.087 s

Press any key to continue.

EXERCISE – 06

Date- 10/02/2022

String handling functions.

Objective-

To understand the string handling functions.

Example-

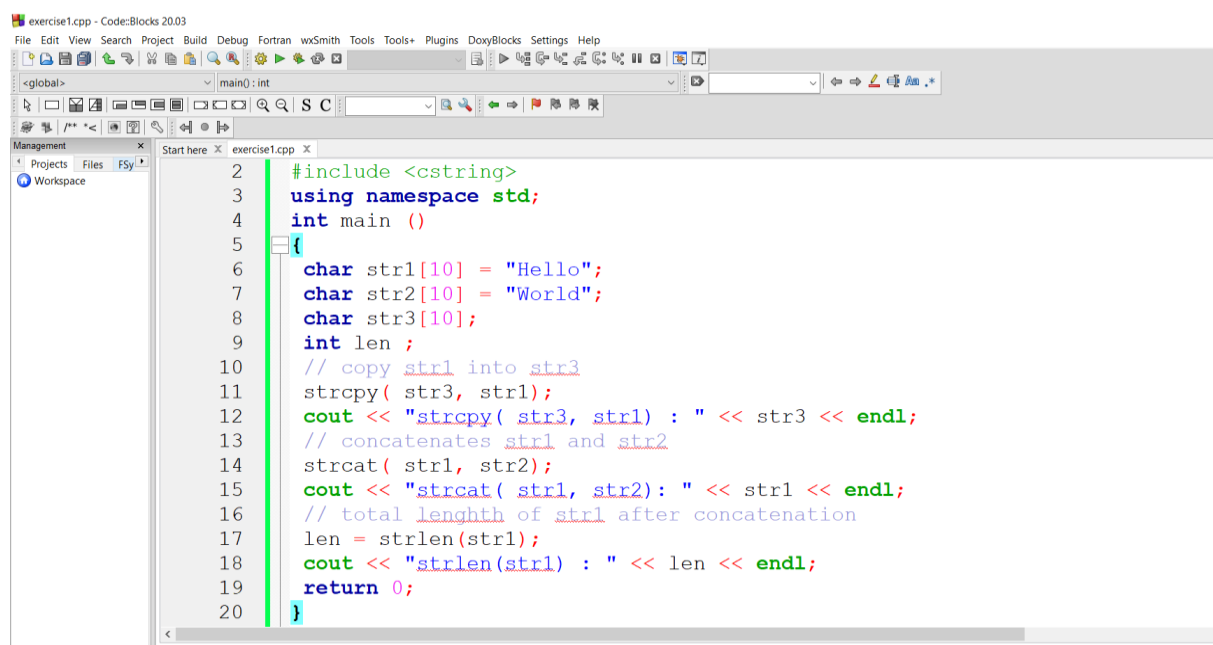
A C++ program to perform basic operations on strings such as copy, concatenate and getting their lengths.

Aim: To make a C++ program to perform basic operations on strings such as copy, concatenate and getting their lengths.

Algorithm-

1. Enter the pre-processor directive “include <iostream>”
2. Write using namespace std;
3. Enter the main()
4. Define strings
5. Copy string 1 into string3
6. Concatenate str1 and str 2
7. Print total length of string after concatenation
8. Exit

Input-



```
exercise1.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DonyBlocks Settings Help
<global> | main0: int
Management | Start here | exercise1.cpp
Projects | Files | FSy
Workspace
2  #include <cstring>
3  using namespace std;
4  int main ()
5  {
6      char str1[10] = "Hello";
7      char str2[10] = "World";
8      char str3[10];
9      int len ;
10     // copy str1 into str3
11     strcpy( str3, str1);
12     cout << "strcpy( str3, str1) : " << str3 << endl;
13     // concatenates str1 and str2
14     strcat( str1, str2);
15     cout << "strcat( str1, str2): " << str1 << endl;
16     // total length of str1 after concatenation
17     len = strlen(str1);
18     cout << "strlen(str1) : " << len << endl;
19     return 0;
20 }
```

Output-

C:\Users\HP\Desktop\exercise1.exe

```
strcpy( str3, str1) : Hello  
strcat( str1, str2): HelloWorld  
strlen(str1) : 10
```

```
Process returned 0 (0x0)   execution time : 0.063 s  
Press any key to continue.
```

EXERCISE – 07

Date- 10/02/2022

Creating and working with classes

Objective-

To understand the working with classes

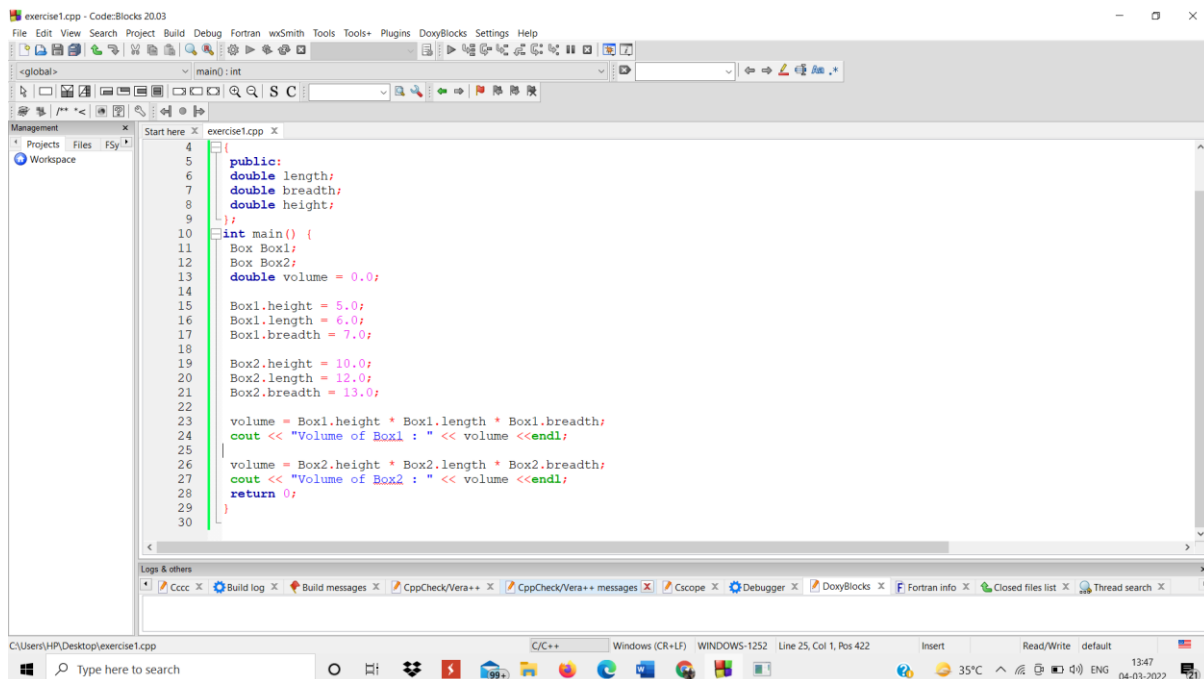
Aim:

To make a C++ program to develop the concept of classes and objects in C++.

Algorithm:

1. Enter the pre-processor directive “include <iostream>”
2. Write using namespace std;
3. Create a class Room
4. Define its members
5. Create object of Room class
6. Assign values to data members
7. Calculate and display the area and volume of the room

Input-

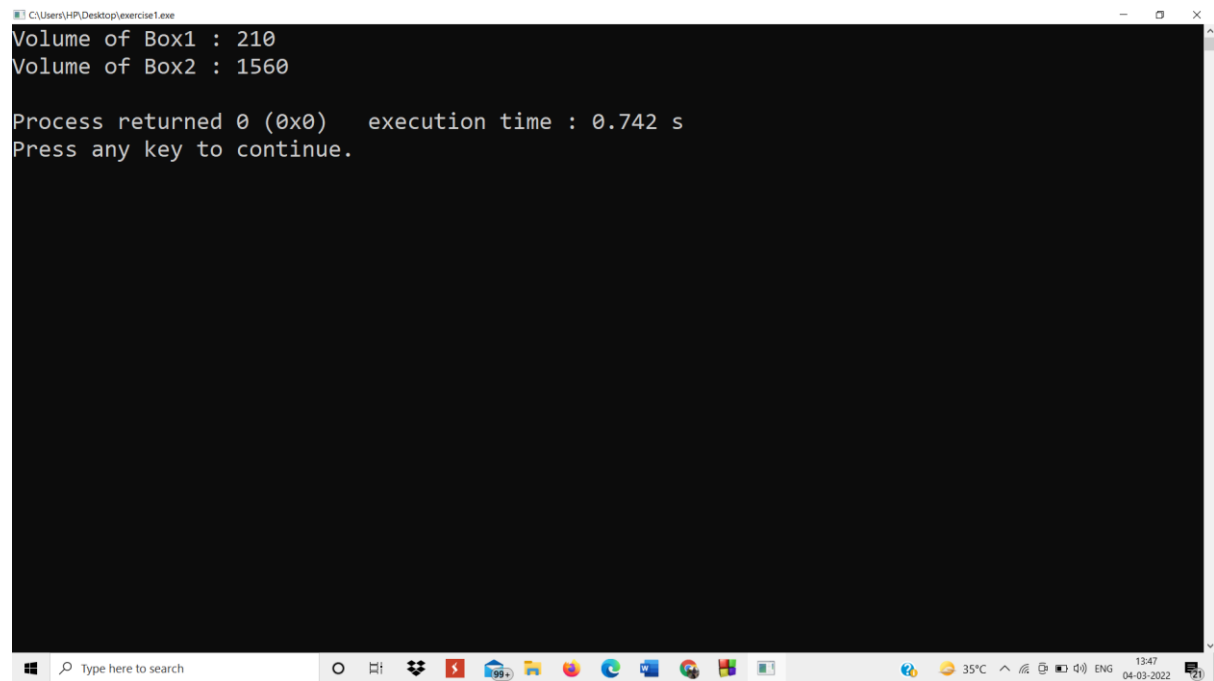


```
1  #include <iostream>
2
3  using namespace std;
4
5  class Box {
6  public:
7      double length;
8      double breadth;
9      double height;
10 };
11
12 int main() {
13     Box Box1;
14     Box Box2;
15     double volume = 0.0;
16
17     Box1.height = 5.0;
18     Box1.length = 6.0;
19     Box1.breadth = 7.0;
20
21     Box2.height = 10.0;
22     Box2.length = 12.0;
23     Box2.breadth = 13.0;
24
25     volume = Box1.height * Box1.length * Box1.breadth;
26     cout << "Volume of Box1 : " << volume << endl;
27
28     volume = Box2.height * Box2.length * Box2.breadth;
29     cout << "Volume of Box2 : " << volume << endl;
30     return 0;
31 }
```


Output-

```
C:\Users\HP\Desktop\exercise1.exe
Volume of Box1 : 210
Volume of Box2 : 1560

Process returned 0 (0x0)   execution time : 0.742 s
Press any key to continue.
```



EXERCISE – 08

Date- 10/02/2022

Illustration of constructors and destructors

Objective-

To understand the working with constructors and destructors

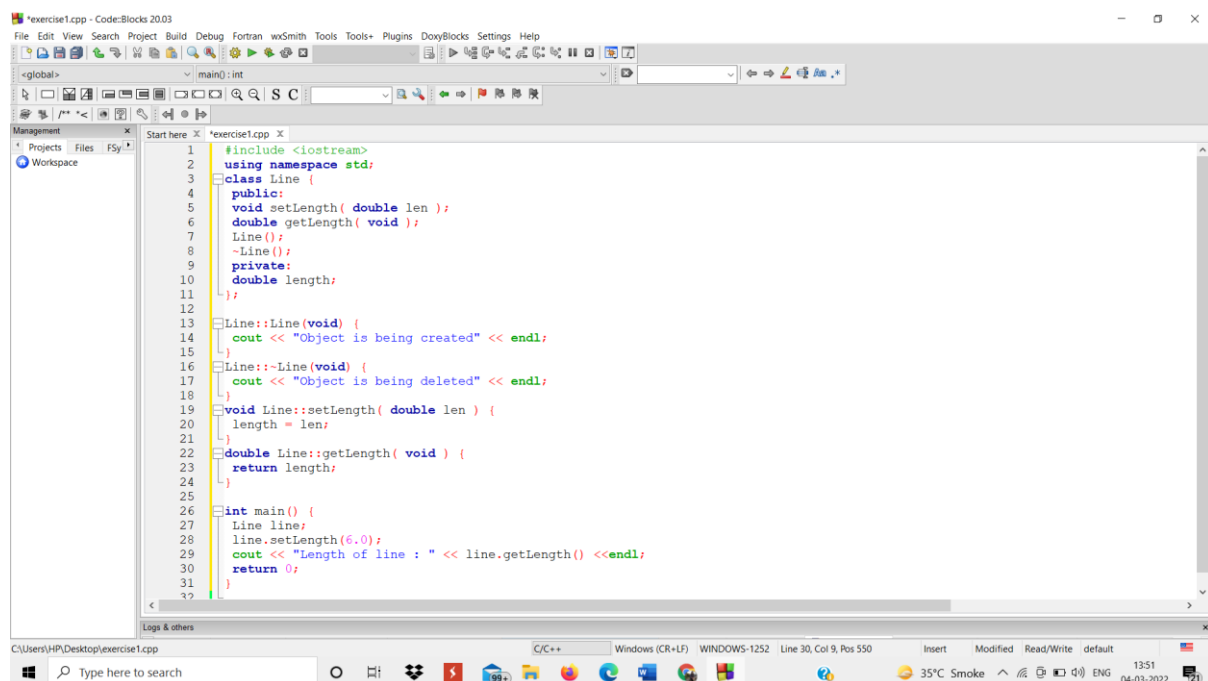
Aim:

To make a C++ program to develop the concept of constructors and destructors

Algorithm:

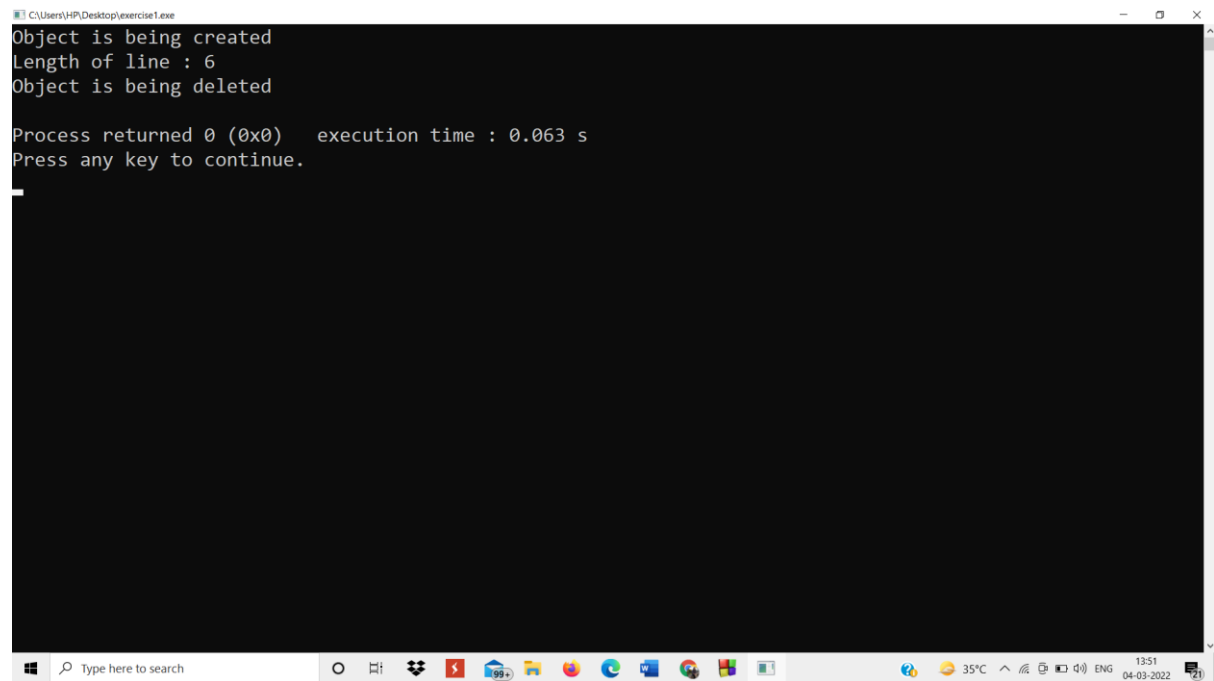
1. Enter the pre-processor directive “include <iostream>”
2. Write using namespace std;
3. Create a class- line
4. Declare its members
5. Declare a constructor ‘line()’
6. Declare a destructor ‘~line()’
7. Write member functions definitions including constructor
8. Write main function for the program

Input-



```
1  #include <iostream>
2  using namespace std;
3  class Line {
4  public:
5      void setLength( double len );
6      double getLength( void );
7      Line();
8      ~Line();
9  private:
10     double length;
11 };
12
13 Line::Line(void) {
14     cout << "Object is being created" << endl;
15 }
16 Line::~Line(void) {
17     cout << "Object is being deleted" << endl;
18 }
19 void Line::setLength( double len ) {
20     length = len;
21 }
22 double Line::getLength( void ) {
23     return length;
24 }
25
26 int main() {
27     Line line;
28     line.setLength(6.0);
29     cout << "Length of line : " << line.getLength() << endl;
30     return 0;
31 }
32
```

OUTPUT-



A screenshot of a Windows command prompt window titled "C:\Users\HP\Desktop\exercice1.exe". The window has a black background with white text. The output text is as follows:

```
Object is being created
Length of line : 6
Object is being deleted

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

The window is positioned over a Windows 10 desktop. The taskbar at the bottom shows the Start button, a search bar with the text "Type here to search", and several pinned application icons including File Explorer, Microsoft Edge, and others. The system tray on the right shows the date and time as "13:51 04-03-2022" and various system icons like network, volume, and battery.

EXERCISE – 09

Date- 10/02/2022

Scope resolution (::) operator

Objective-

To understand the working with constructors and destructors

Example-

Program to define the member function outside of the class using the scope resolution (::) operator

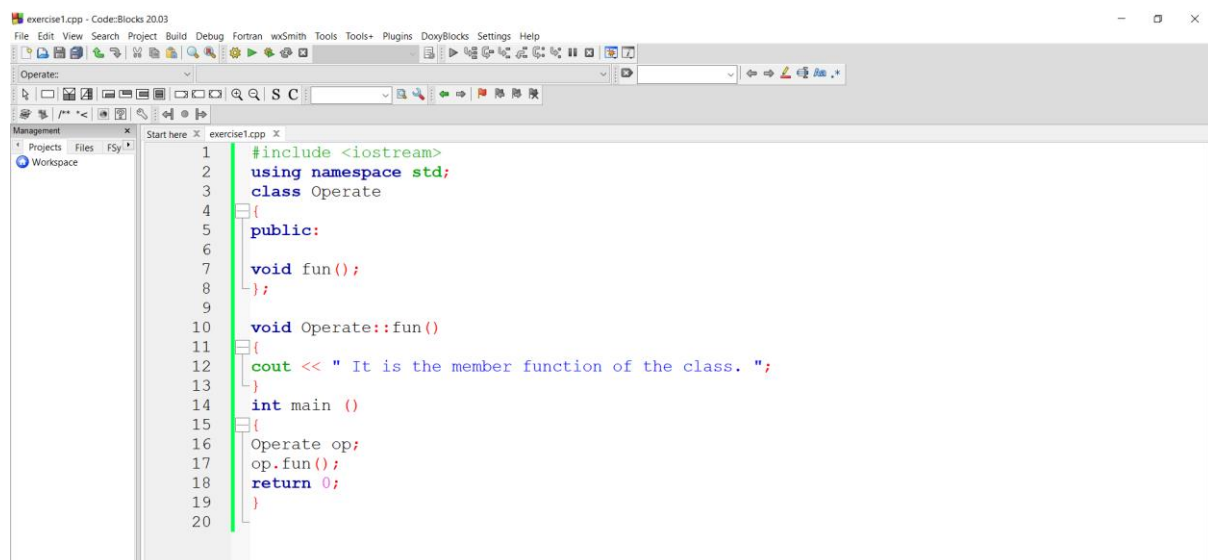
Aim:

To make a C++ program to develop the concept of scope resolution (::) operator.

Algorithm:

1. Enter the pre-processor directive “include <iostream>”
2. Write using namespace std;
3. Create a class operate
4. Write declaration of the member function
5. Define the member function outside the class
6. return_type Class_Name::function_name
7. create an object of the class Operate
8. Exit


Input-



```
1 #include <iostream>
2 using namespace std;
3 class Operate
4 {
5 public:
6
7 void fun();
8 };
9
10 void Operate::fun()
11 {
12 cout << " It is the member function of the class. ";
13 }
14 int main ()
15 {
16 Operate op;
17 op.fun();
18 return 0;
19 }
20
```

OUTPUT:

```
C:\Users\HP\Desktop\exercise1.exe
It is the member function of the class.
Process returned 0 (0x0)   execution time : 0.071 s
Press any key to continue.
```

The image shows a Windows taskbar at the bottom of the screen. It includes the Start button, a search bar with the text "Type here to search", and a taskbar with several pinned application icons. On the right side of the taskbar, there are system icons for network, volume, and battery, along with the date and time "13:56 04-03-2022".