

PRACTICAL FILE ON OBJECT ORIENTED PROGRAMMING USING C++



**SESSION 2024 – 2025
DAYANAND COLLEGE
HISAR**

Submitted to:-

Mrs. Pushpa Mam

[Assistant Professor
Computer Science
Department]

Submitted by:-

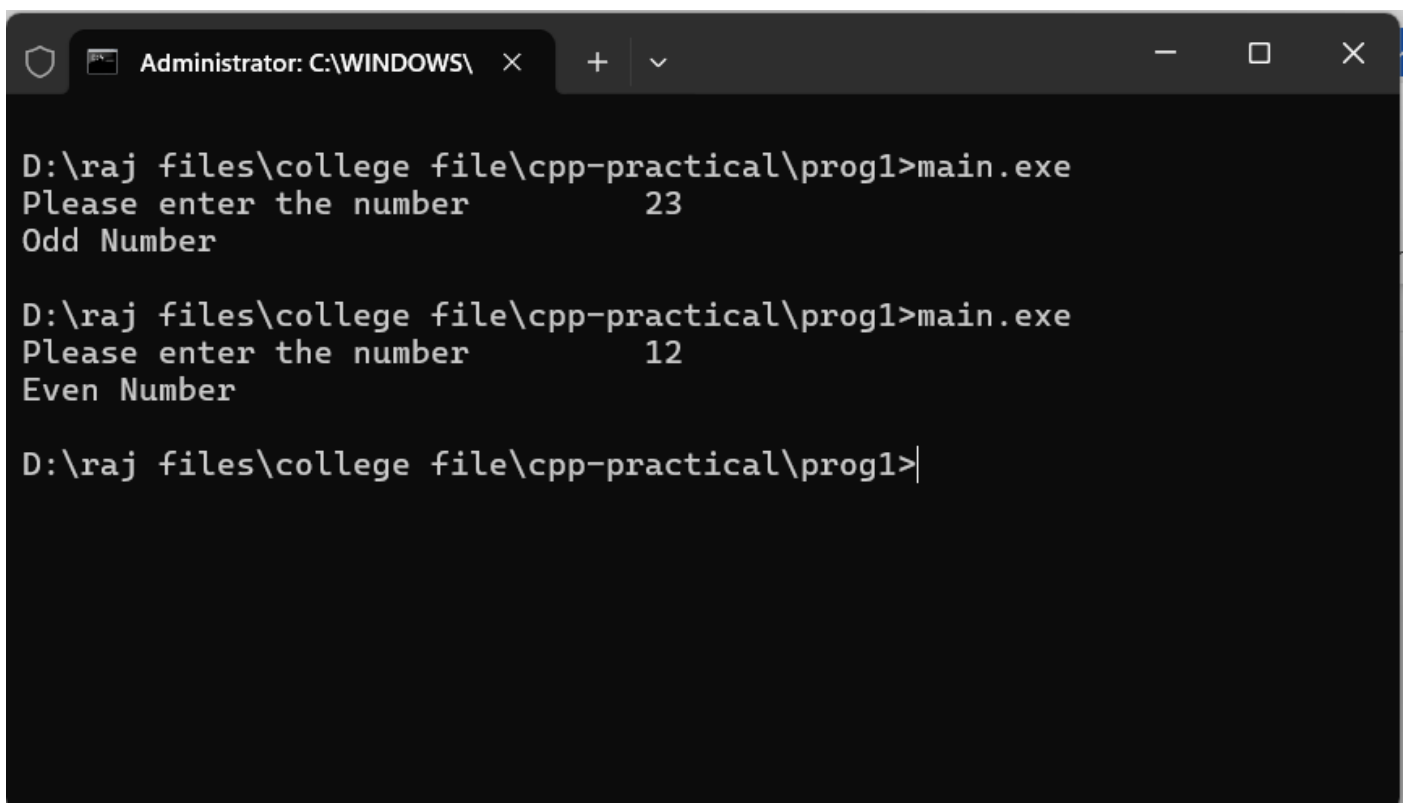
Sahil

Class:- BCA-2nd Sem.

Uni. ROLL NO.: 243042220142

Index

| | |
|---|----|
| Question 1. Write a program to check whether a number is even or not..... | 4 |
| Question 2. Write a program to check whether a number is prime or not..... | 6 |
| Question 3. Write a program to find the factorial of given number. | 8 |
| Question 4. Write a program to find out the area of square, rectangle and circle by using function overloading. | 10 |
| Question 5. Write a program find out the average of two members by using member function define inside the class definition. | 14 |
| Question 6. Write a program to find the simple interest by using member member function define outside the class definition. | 16 |
| Question 7. Write a program to perform all arithmetic operation such as addition, subtraction, multiplication, division and modulus using inline function. | 20 |
| Question 8. Write a program to explain the concept of static data member. | 24 |
| Question 9. Write a program to explain the concept of static member function. | 26 |
| Question 10. Write a program to use the concept of friend function. | 28 |
| Question 11. Write a program to use the concept of friend class..... | 30 |
| Question 12. Write a program to use the concept of parameterized constructor. | 32 |
| Question 13. Write a program to use the concept of copy constructor. | 36 |
| Question 14. Write a program to use the concept of constructor overloading. | 38 |
| Question 15. Write a program to use the concept of destructor. | 40 |
| Question 16. Write a program to overload the unary operator by using member function..... | 42 |
| | 43 |
| Question 17. Write a program to overload the binary operator by using friend function..... | 44 |
| Question 18. Write a program to use the concept of single inheritance (by using public method). | 48 |
| | 51 |
| Question 19. Write a program to use the concept of single inheritance (by using private method). | 52 |
| | 53 |
| Question 20. Write a program to sue the concept of multilevel inheritance (by using protected method). . | 54 |
| Question 21. Write a program by using concept of multiple inheritance. | 58 |
| Question 22. Write a program by using various Manipulators..... | 60 |
| Question 23. Write a program to use the concept of function overriding. | 62 |
| Question 24. Write a program to use the concept of virtual base class..... | 64 |
| Question 25. Write a program to use the concept of inheritance and parameterized constructor. | 66 |
| Question 26. Write a program to use the concept of virtual function. | 68 |



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following sequence of commands and outputs:

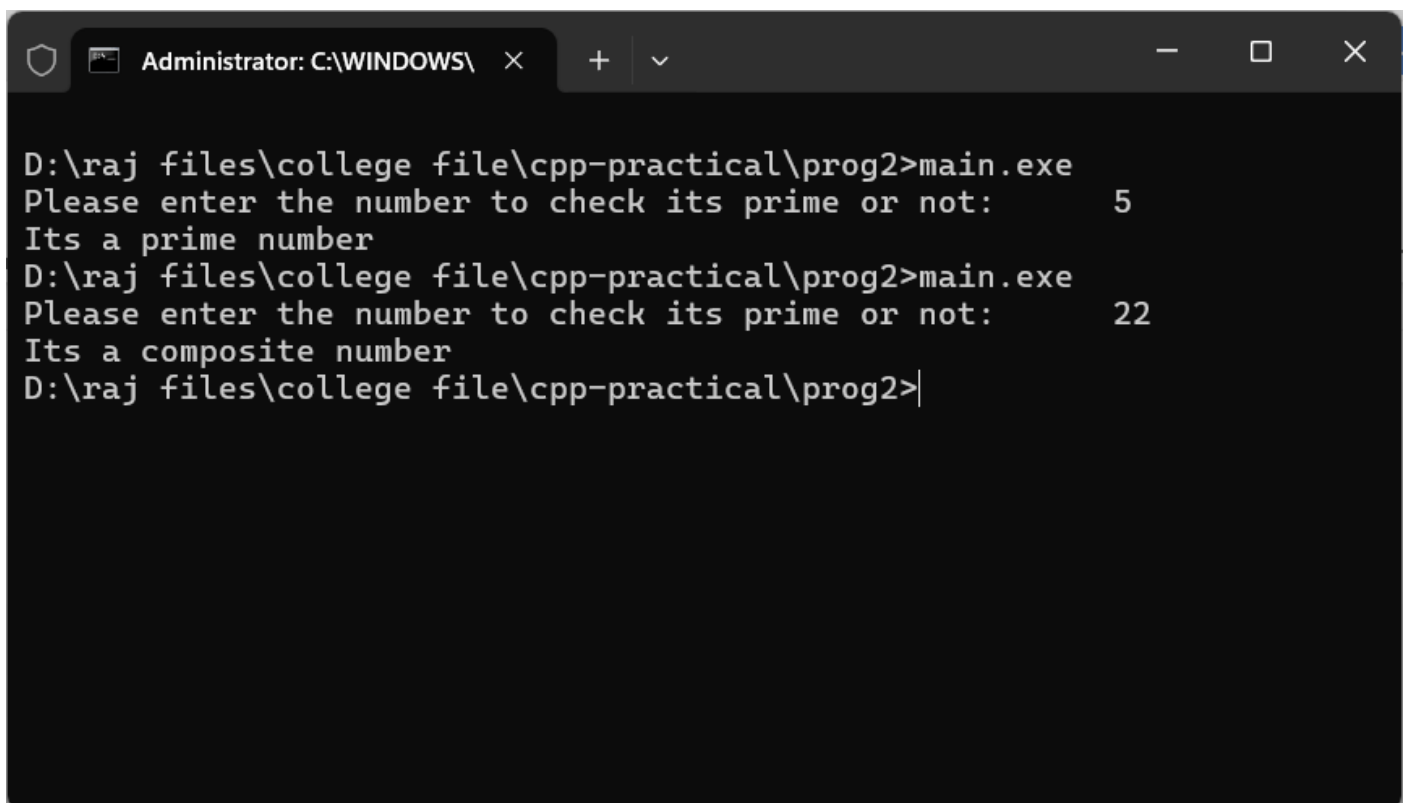
```
D:\raj files\college file\cpp-practical\prog1>main.exe
Please enter the number      23
Odd Number

D:\raj files\college file\cpp-practical\prog1>main.exe
Please enter the number      12
Even Number

D:\raj files\college file\cpp-practical\prog1>|
```

Question 1. Write a program to check whether a number is even or not.

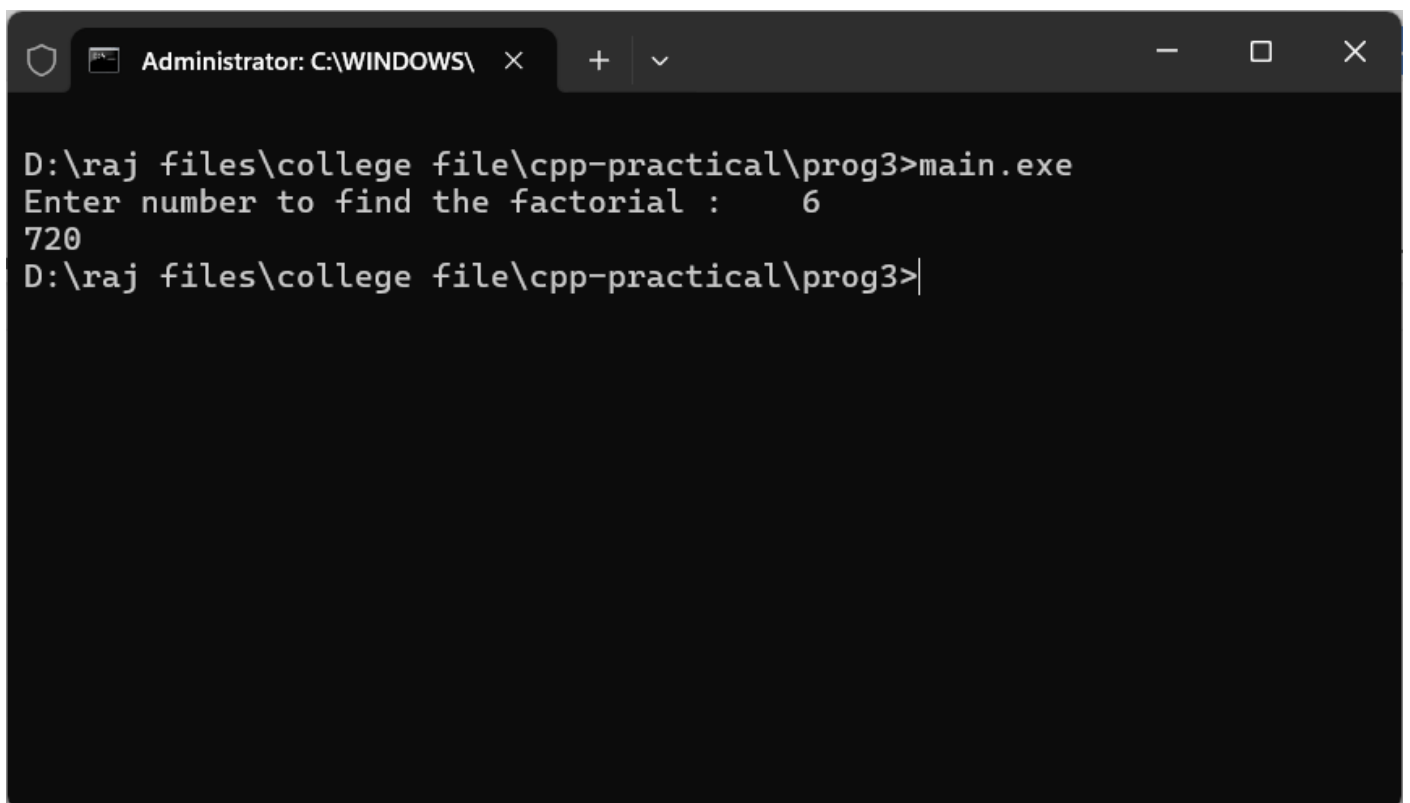
```
#include <iostream>
using namespace std;
void parity(int num);
int main()
{
    int num;
    cout << "Please enter the number \t";
    cin >> num;
    parity(num);
    return 0;
}
void parity(int num)
{
    (num%2 == 0)? cout << "Even Number\n" : cout << "Odd
Number\n";
}
```



```
D:\raj files\college file\cpp-practical\prog2>main.exe
Please enter the number to check its prime or not:      5
Its a prime number
D:\raj files\college file\cpp-practical\prog2>main.exe
Please enter the number to check its prime or not:      22
Its a composite number
D:\raj files\college file\cpp-practical\prog2>|
```

Question 2. Write a program to check whether a number is prime or not.

```
#include<iostream>
using namespace std;
void primality(int num);
int main()
{
    int num;
    cout << "Please enter the number to check its prime
or not:\t";
    cin >> num;
    primality(num);
    return 0;
}
void primality(int num)
{
    int flag = 0;
    if (num<=1){cout << "Its not a prime number or
composite number"; return ;}
    for (int i = 2; i<num; i++)
    {
        if ((num % i) == 0){flag++; break;}
    }
    if (flag) {cout <<"Its a composite number"; return
;}}
    else {cout <<"Its a prime number"; return ;}}
```



A screenshot of a Windows command prompt window. The title bar shows 'Administrator: C:\WINDOWS\' with standard window controls. The command prompt displays the following text:

```
D:\raj files\college file\cpp-practical\prog3>main.exe
Enter number to find the factorial :    6
720
D:\raj files\college file\cpp-practical\prog3>|
```


Question 3. Write a program to find the factorial of given number.

```
#include<iostream>
using namespace std;
int fact(int num);
int main()
{
    int num;
    cout << "Enter number to find the factorial : \t";
    cin >> num;
    cout << fact(num);
    return 0;
}
int fact(int num)
{
    if((num == 0) || (num == 1)){return num;}
    else {return num * fact(num-1);}
}
```

```
Administrator: C:\WINDOWS\ × + ∨ - □ ×

D:\raj files\college file\cpp-practical\prog4>main.exe
Enter shape number whose area to find
1. Square      2. Rectangle  3. Circle

1

Please enter the length of side of Sqaure :    12
144

D:\raj files\college file\cpp-practical\prog4>main.exe
Enter shape number whose area to find
1. Square      2. Rectangle  3. Circle

2

Please enter the length of Rectangle :  12 23
Please enter the breadth of Rectangle : 276

D:\raj files\college file\cpp-practical\prog4>main.exe
Enter shape number whose area to find
1. Square      2. Rectangle  3. Circle

3

Please enter the radius of circle :    13
530.66

D:\raj files\college file\cpp-practical\prog4>
```

Question 4. Write a program to find out the area of square, rectangle and circle by using function overloading.

```
#include <iostream>
#define PI 3.14
using namespace std;
void area(int length);
void area(int length, int breadth);
void area(int radius, char circle);
int main()
{
    int opt, length, breadth, radius;
    cout<<"Enter shape number whose area to find\n1. Square\t2.
Rectangle\t3. Circle\n"<<endl;
    cin >> opt;
    switch (opt)
    {
        case 1:
            cout << "\nPlease enter the length of side of
Sqaure :\t";
            cin >> length;
            area(length);
            break;
        case 2:
            cout << "\nPlease enter the length of Rectangle
:\t";
            cin >> length;
            cout << "Please enter the breadth of Rectangle
:\t";
            cin >> breadth;
```



```

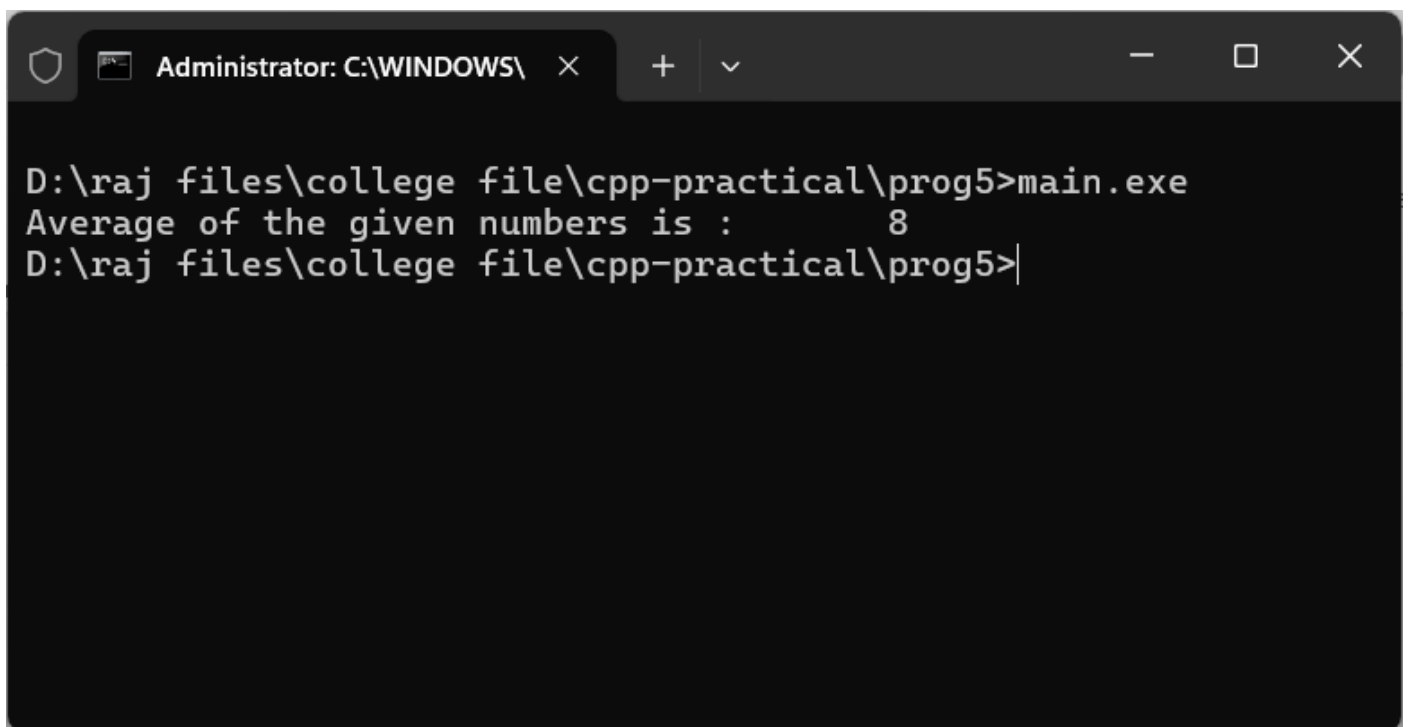
        area(length, breadth);
        break;
    case 3:
        cout << "\nPlease enter the radius of circle :\t";
        cin >> radius;
        area(radius, 'c');
        break;
    }
    return 0;
}

void area(int length)
{
    cout << length*length << endl;
}

void area(int length, int breadth)
{
    cout << length*breadth << endl;
}

void area(int radius, char circle)
{
    cout << (PI * radius * radius) << endl;
}

```

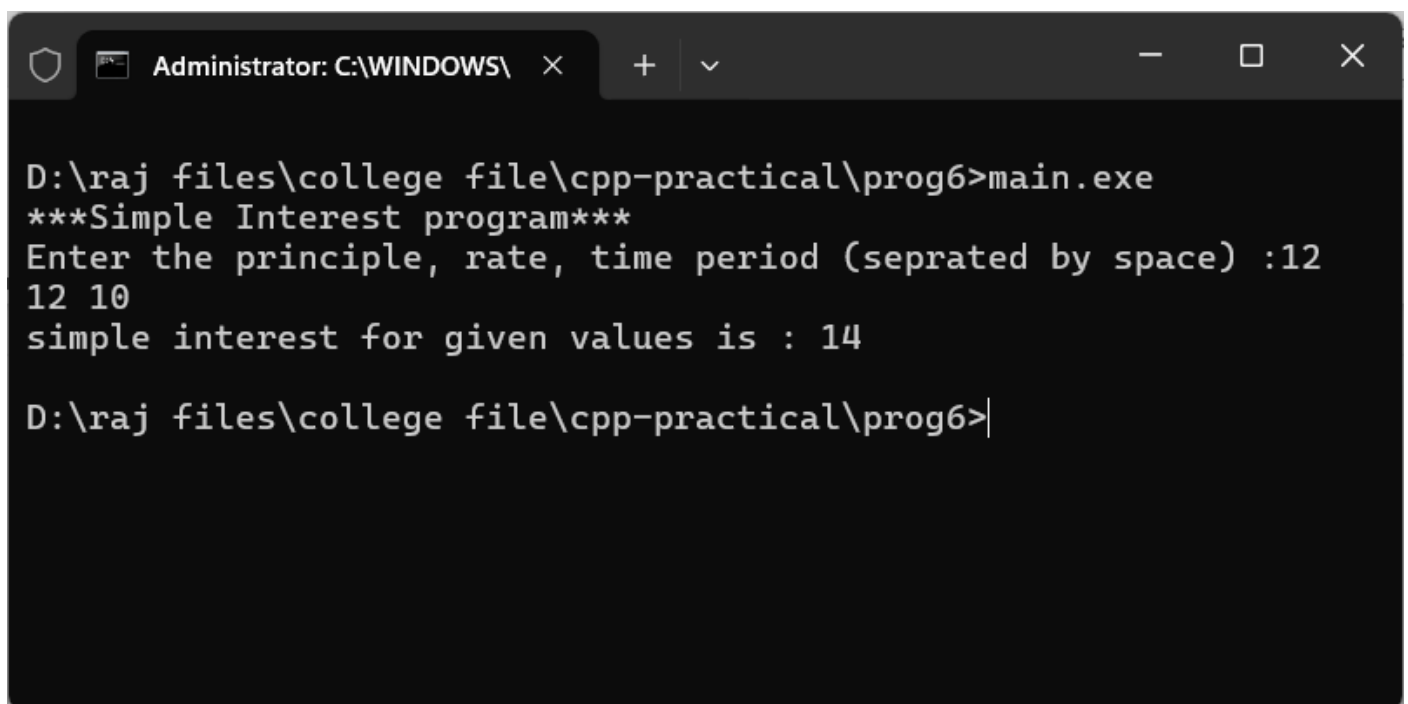


A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt area has a black background with white text. The first line shows the directory path "D:\raj files\college file\cpp-practical\prog5" followed by a prompt character ">" and the command "main.exe". The second line shows the output "Average of the given numbers is : 8". The third line shows the directory path "D:\raj files\college file\cpp-practical\prog5" followed by a prompt character ">" and a vertical cursor.

```
D:\raj files\college file\cpp-practical\prog5>main.exe
Average of the given numbers is :      8
D:\raj files\college file\cpp-practical\prog5>|
```

Question 5. Write a program find out the average of two members by using member function define inside the class definition.

```
#include<iostream>
using namespace std;
class Math_Function
{
    int num1, num2;
public:
    Math_Function(int first, int second)
    {
        num1 = first;
        num2 = second;
    }
    void average()
    {
        cout<< "Average of the given numbers is :\t"<<
((num1 + num2)/2);
    }
};
int main()
{
    Math_Function obj1(5, 12);
    obj1.average();
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows 'Administrator: C:\WINDOWS\' with standard window controls. The command prompt displays the execution of 'main.exe' in the directory 'D:\raj files\college file\cpp-practical\prog6'. The program outputs a title, prompts for input, receives '12 10', and displays the result '14'.

```
D:\raj files\college file\cpp-practical\prog6>main.exe
***Simple Interest program***
Enter the principle, rate, time period (seprated by space) :12
12 10
simple interest for given values is : 14
D:\raj files\college file\cpp-practical\prog6>
```


Question 6. Write a program to find the simple interest by using member member function define outside the class definition.

```
#include<iostream>
using namespace std;
class MyClass
{
    int principle, rate, time;
public:
    void SetValues(int p, int r, int t);
    void Interest();
};
void MyClass :: SetValues(int p, int r, int t)
{
    principle = p; rate = r; time = t;
}
void MyClass :: Interest()
{
    cout << "simple interest for given values is : " <<
((principle * rate * time) / 100) << endl;
}
int main()
{
    int p, r, t;
    MyClass obj1;
    cout << "****Simple Interest program****" << endl;
    cout << "Enter the principle, rate, time period (separated
by space) : ";
    cin >> p >> r >> t;
```



```
    obj1.SetValues(p, r, t);  
    obj1.Interest();  
    return 0;  
}
```

```
Administrator: C:\WINDOWS\ ... + v
D:\raj files\college file\cpp-practical\prog7>main.exe
Which operation to perform(Option number) :
1. Addition    2. Subraction    3. Multiplication    4. Division    5. modulus    : 1
Enter number 1: 12
Enter number 2: 13
25

D:\raj files\college file\cpp-practical\prog7>main.exe
Which operation to perform(Option number) :
1. Addition    2. Subraction    3. Multiplication    4. Division    5. modulus    : 2
Enter number 1: 15
Enter number 2: 10
5

D:\raj files\college file\cpp-practical\prog7>main.exe
Which operation to perform(Option number) :
1. Addition    2. Subraction    3. Multiplication    4. Division    5. modulus    : 3
Enter number 1: 12
Enter number 2: 16
192

D:\raj files\college file\cpp-practical\prog7>main.exe
Which operation to perform(Option number) :
1. Addition    2. Subraction    3. Multiplication    4. Division    5. modulus    : 4
Enter number 1: 20
Enter number 2: 4
5

D:\raj files\college file\cpp-practical\prog7>main.exe
Which operation to perform(Option number) :
1. Addition    2. Subraction    3. Multiplication    4. Division    5. modulus    : 5
Enter number 1: 20
Enter number 2: 4
0

D:\raj files\college file\cpp-practical\prog7>|
```

Question 7. Write a program to perform all arithmetic operation such as addition, subtraction, multiplication, division and modulus using inline function.

```
#include<iostream>

using namespace std;

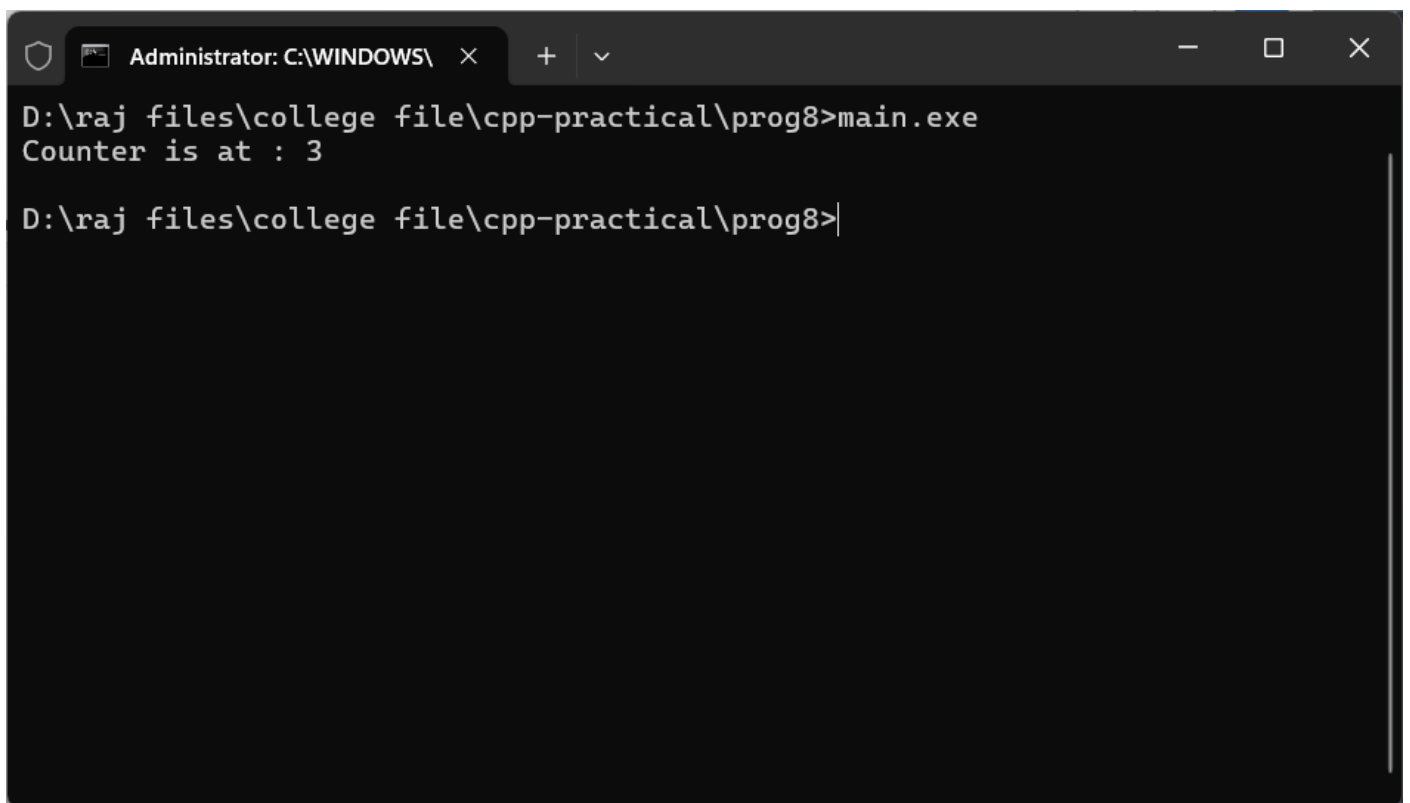
int num1, num2;

inline void add(){cout << num1+num2 << endl;}
inline void sub(){cout << num1-num2 << endl;}
inline void mult(){cout << num1*num2 << endl;}
inline void div(){cout << num1/num2 << endl;}
inline void mod(){cout << num1%num2 << endl;}

int main()
{
    int opt;
    cout << "Which operation to perform(Option number) :\n";
    cout << "1. Addition\t2. Subraction\t3. Multiplication\t4.
Division\t5. modulus\t: ";
    cin >> opt;
    cout << "Enter number 1: ";
    cin >> num1;
    cout << "Enter number 2: ";
    cin >> num2;
    switch (opt)
    {
        case 1: add(); break;
        case 2: sub(); break;
        case 3: mult(); break;
        case 4: div(); break;
```



```
        case 5: mod(); break;
        default:
            cout << "Invalid input";
    }
    return 0;
}
```

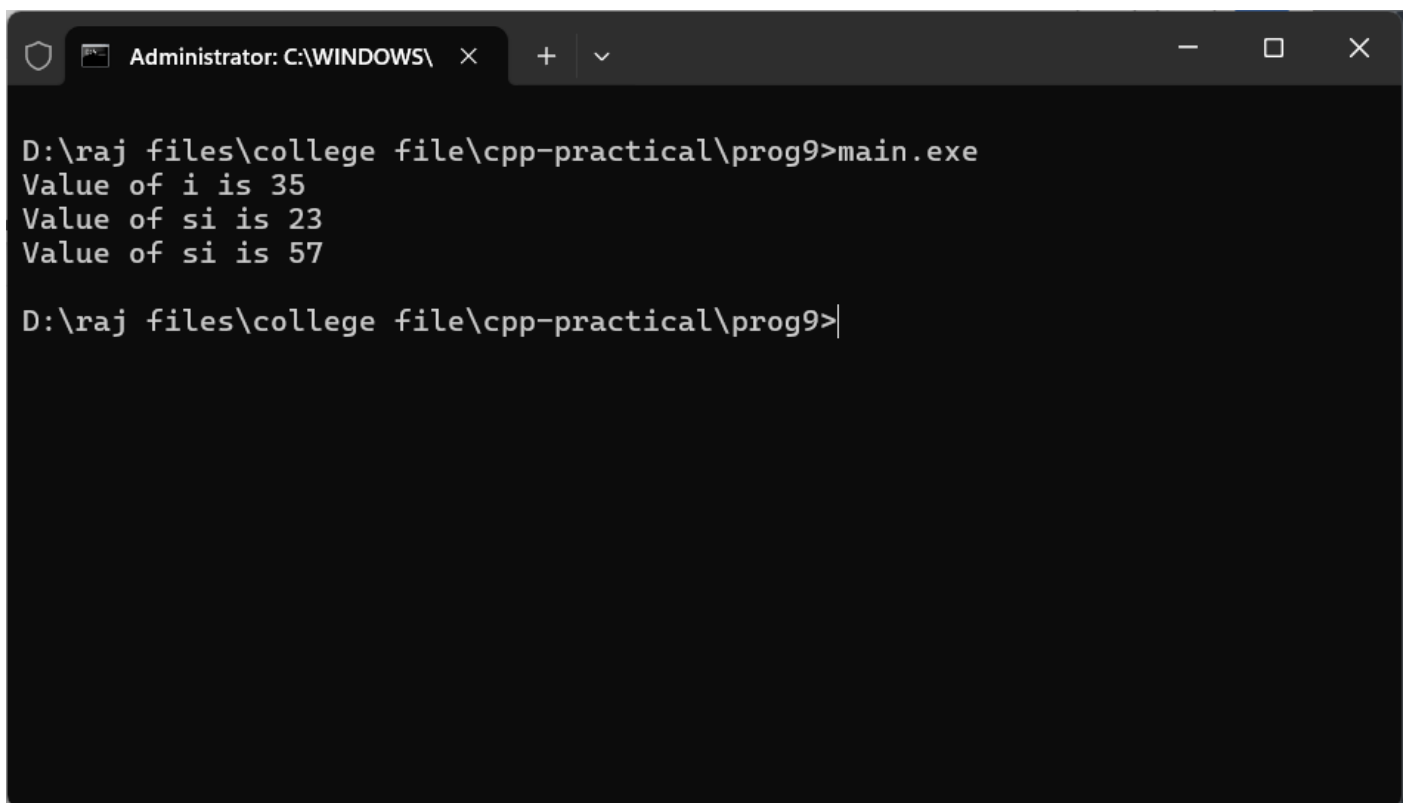


A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt itself has a black background with white text. The first line shows the command `D:\raj files\college file\cpp-practical\prog8>main.exe` and the output `Counter is at : 3`. The second line shows the prompt `D:\raj files\college file\cpp-practical\prog8>` with a cursor at the end.

```
Administrator: C:\WINDOWS\ × + ▾  
D:\raj files\college file\cpp-practical\prog8>main.exe  
Counter is at : 3  
D:\raj files\college file\cpp-practical\prog8>|
```


Question 8. Write a program to explain the concept of static data member.

```
#include<iostream>
using namespace std;
class Static_Data
{
    public:
        static int counter;
        Static_Data(){counter++;}
};
int Static_Data::counter = 0;
int main()
{
    Static_Data obj1, obj2, obj3;
    cout << "Counter is at :\t" << Static_Data::counter<<endl;
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by a close button (X). To the right of the title bar are standard window controls: a plus sign, a dropdown arrow, a minus sign, a maximize button (square), and a close button (X). The command prompt area is black with white text. The first line shows the directory path "D:\raj files\college file\cpp-practical\prog9" followed by a prompt character ">" and the command "main.exe". The next three lines show the output of the program: "Value of i is 35", "Value of si is 23", and "Value of si is 57". The final line shows the directory path "D:\raj files\college file\cpp-practical\prog9" followed by a prompt character ">" and a vertical bar "|", indicating the cursor is ready for the next command.

```
D:\raj files\college file\cpp-practical\prog9>main.exe
Value of i is 35
Value of si is 23
Value of si is 57

D:\raj files\college file\cpp-practical\prog9>|
```

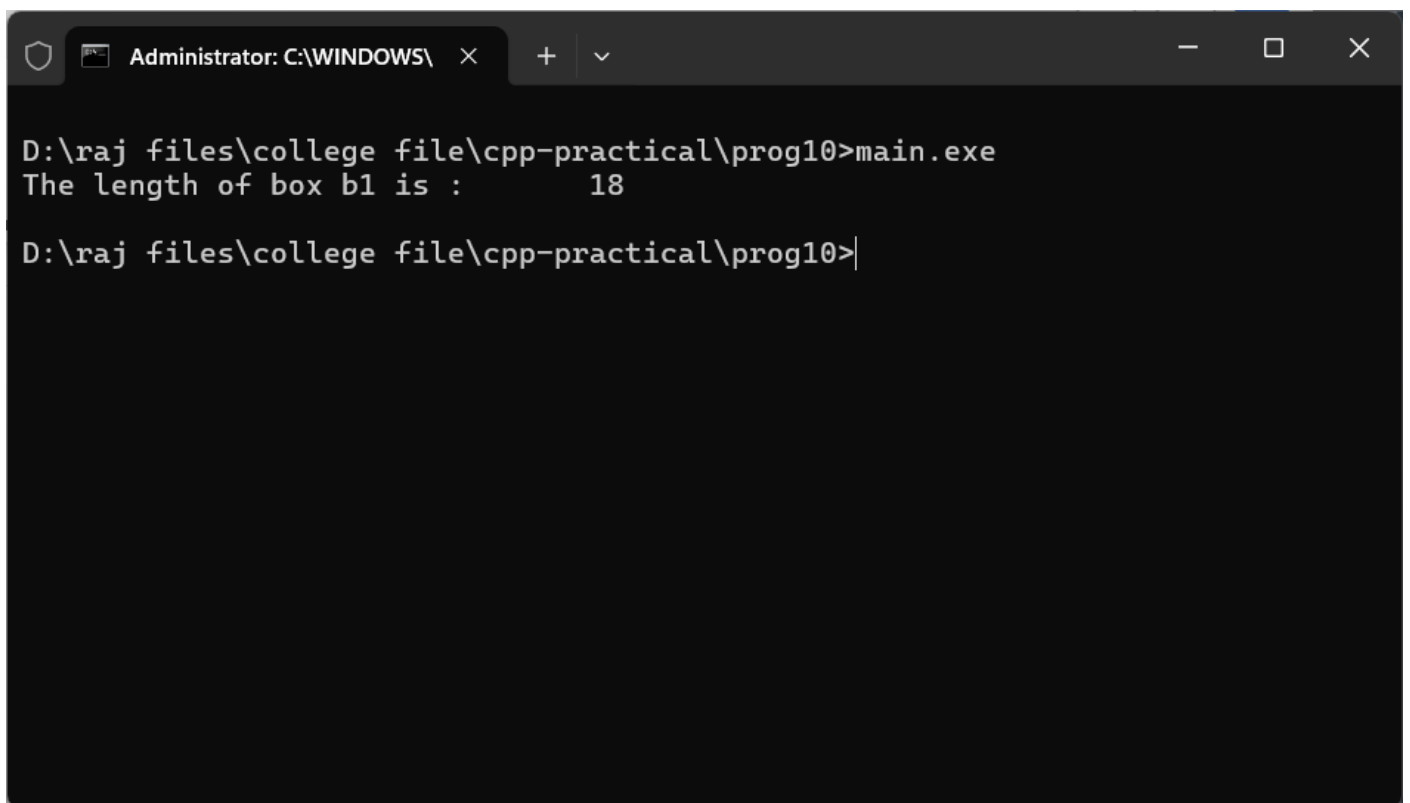
Question 9. Write a program to explain the concept of static member function.

```
#include<iostream>
using namespace std;
class MyClass
{
    private:
        int i;
        static int si;
    public:
        void set_i(int arg){i = arg;}
        static void set_si(int arg){si = arg;}

        void print_i(){cout << "Value of i is " << i << endl;}
        static void print_si(){cout << "Value of si is " << si
<< endl;}
};
int MyClass :: si = 23;
int main()
{
    MyClass obj1;
    obj1.set_i(35);
    obj1.print_i();

    MyClass::print_si();
    MyClass::set_si(57);
    MyClass::print_si();
    return 0;
}
```

}

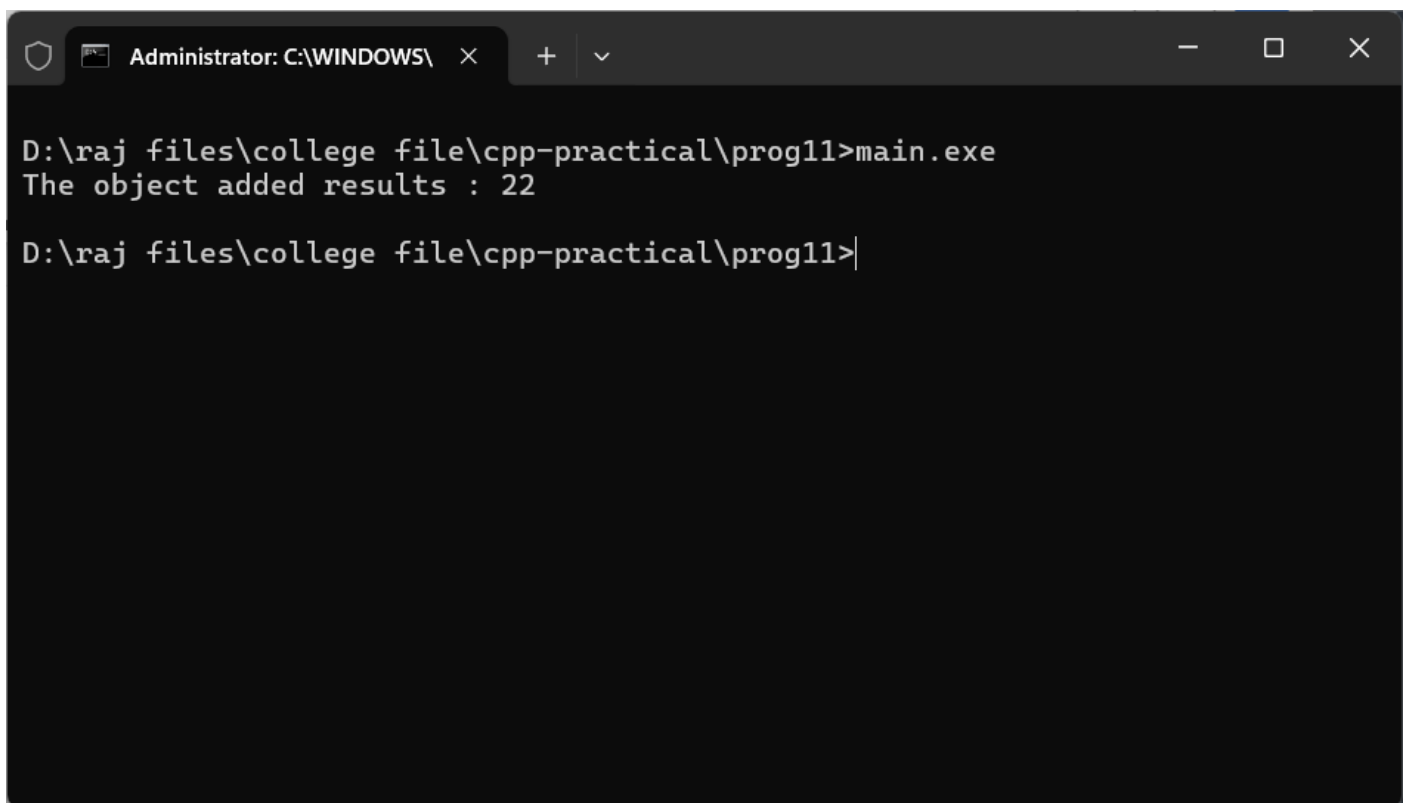


A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt itself has a black background with white text. The first line shows the command `D:\raj files\college file\cpp-practical\prog10>main.exe`. The second line shows the output `The length of box b1 is : 18`. The third line shows the prompt `D:\raj files\college file\cpp-practical\prog10>` with a cursor at the end.

```
Administrator: C:\WINDOWS\ × + ∨ — □ ×  
D:\raj files\college file\cpp-practical\prog10>main.exe  
The length of box b1 is : 18  
D:\raj files\college file\cpp-practical\prog10>|
```

Question 10. Write a program to use the concept of friend function.

```
#include<iostream>
using namespace std;
class Box
{
    int length;
public:
    Box(int len){length = len;}
    friend int print_len(Box);
};
int print_len(Box b){b.length += 10; return b.length;}
int main()
{
    Box b1(8);
    cout << "The length of box b1 is :\t" << print_len(b1) <<
endl;
    return 0;
}
```

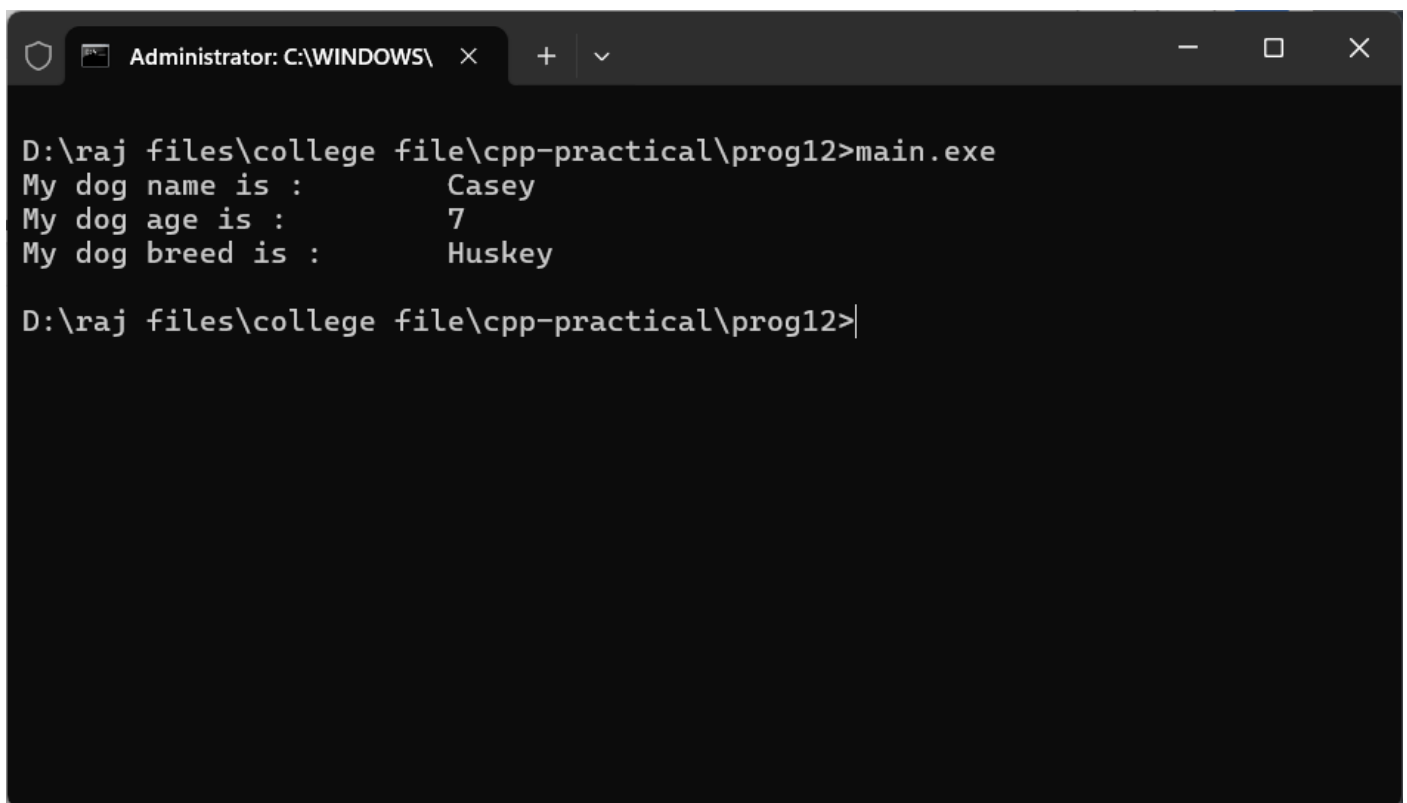


A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by a close button (X). Below the title bar, the command prompt displays the following text:

```
D:\raj files\college file\cpp-practical\prog11>main.exe
The object added results : 22
D:\raj files\college file\cpp-practical\prog11>|
```

Question 11. Write a program to use the concept of friend class.

```
#include<iostream>
using namespace std;
class Friend_Class;
class Base_Class
{
    int size1;
    friend Friend_Class;
public:
    Base_Class():size1(12){}
};
class Friend_Class
{
    int size2;
public:
    Friend_Class():size2(10){}
    int add(){Base_Class b1; return b1.size1 + size2;}
};
int main()
{
    Friend_Class obj1;
    cout << "The object added results : " << obj1.add() <<
endl;
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt itself has a black background with white text. The first line shows the directory path "D:\raj files\college file\cpp-practical\prog12" followed by a prompt character ">" and the command "main.exe". The next three lines show the output of the program: "My dog name is : Casey", "My dog age is : 7", and "My dog breed is : Huskey". The final line shows the directory path followed by a prompt character ">" and a cursor.

```
D:\raj files\college file\cpp-practical\prog12>main.exe
My dog name is :      Casey
My dog age is :       7
My dog breed is :     Huskey

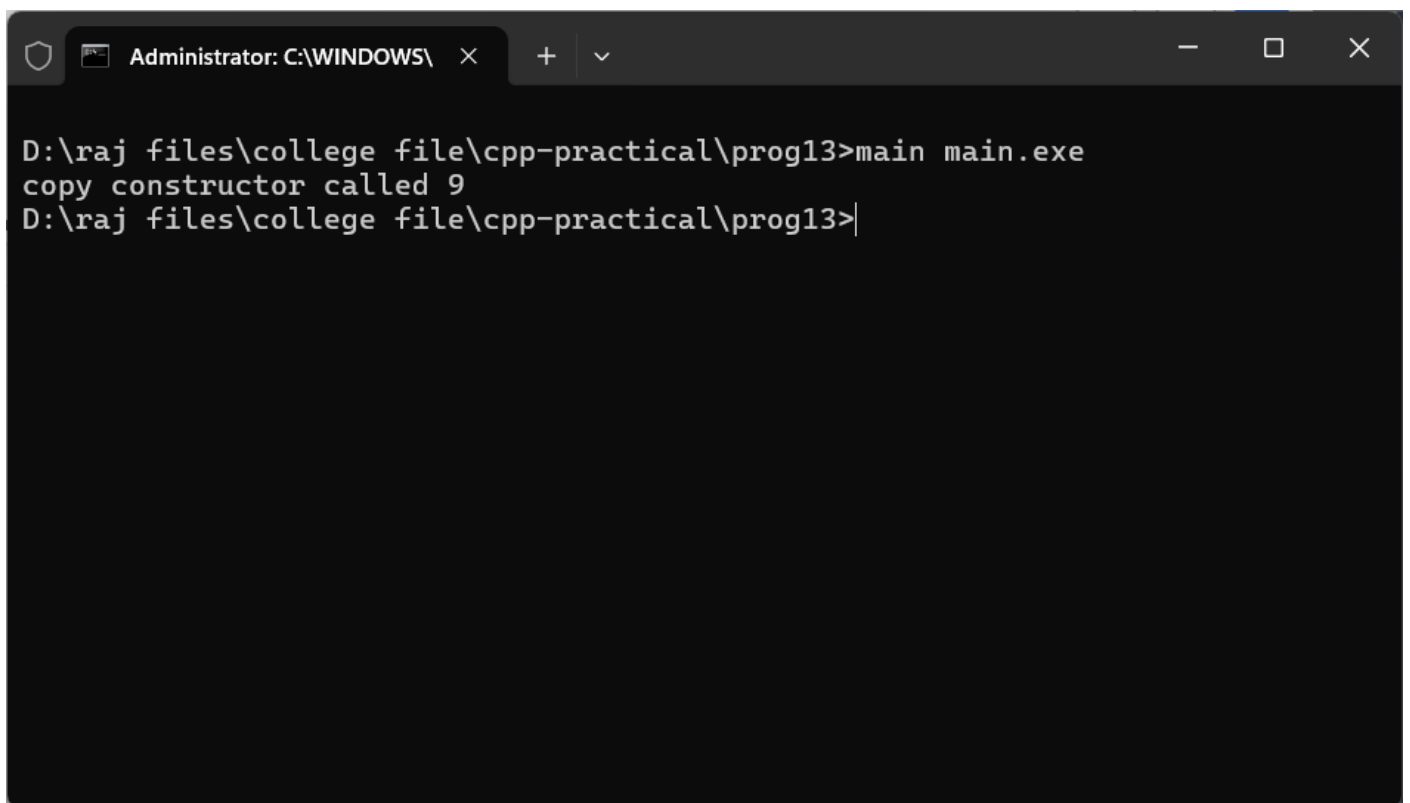
D:\raj files\college file\cpp-practical\prog12>|
```


Question 12. Write a program to use the concept of parameterized constructor.

```
#include<iostream>
#include<string>
using namespace std;
class Dog
{
    private:
        string name;
        int age;
        string breed;
    public:
        Dog(string dog_name, int dog_age, string dog_breed)
        {
            name = dog_name;
            age = dog_age;
            breed = dog_breed;
        }
        void dog_info()
        {
            cout << "My dog name is : \t" << name << endl;
            cout << "My dog age is : \t" << age << endl;
            cout << "My dog breed is :\t" << breed << endl;
        }
};
int main()
{
```



```
Dog my_dog("Casey", 7, "Huskey");  
    my_dog.dog_info();  
    return 0;  
}
```



```
D:\raj files\college file\cpp-practical\prog13>main main.exe
copy constructor called 9
D:\raj files\college file\cpp-practical\prog13>
```

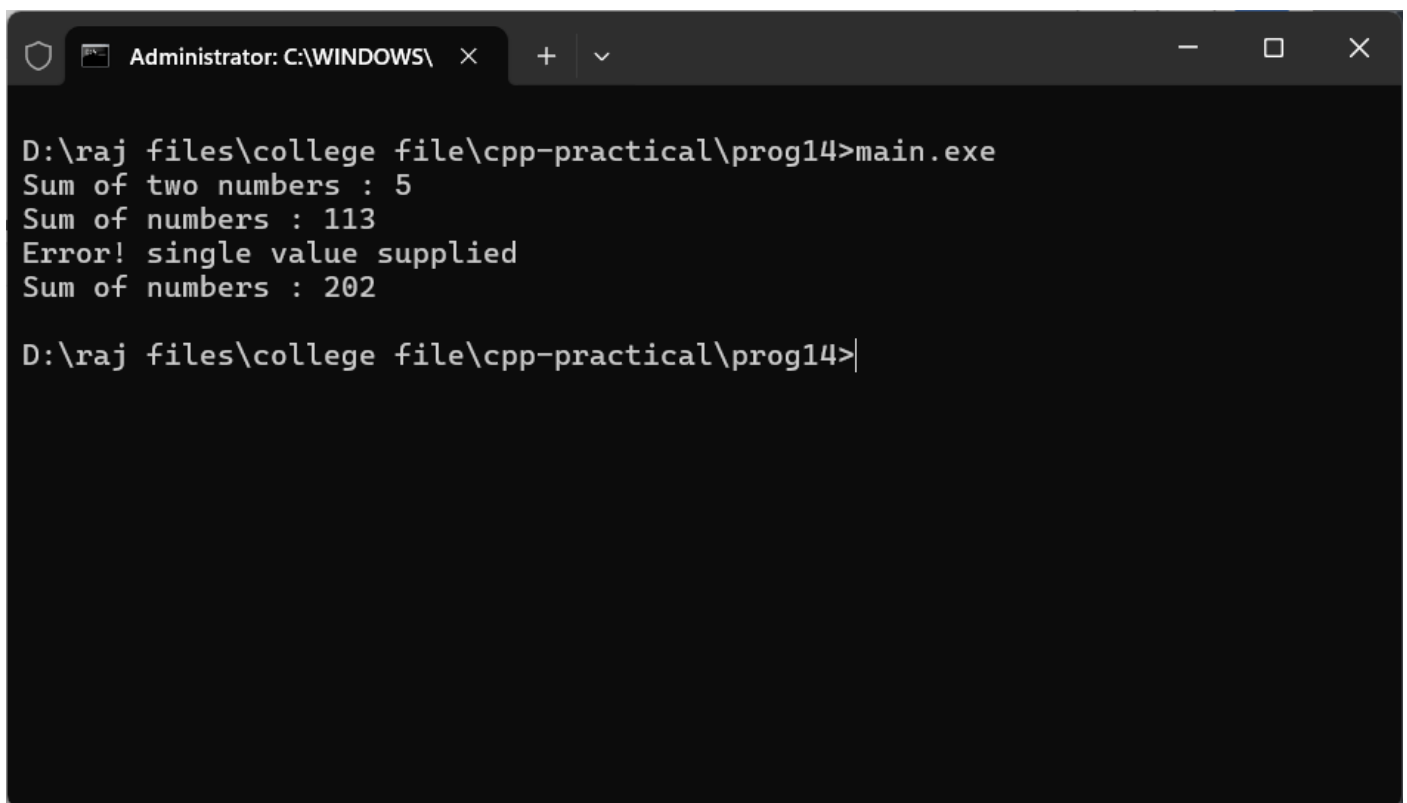
Question 13. Write a program to use the concept of copy constructor.

```
#include <iostream>

using namespace std;

class sample{
public:
    int x,y;
    sample(int a,int b):x(a),y(b){}
    sample(sample &obj){
        x=obj.x;
        y=obj.y;
        cout<<"copy constructor called "<<x+y;
    }
};

int main()
{
    sample s1(4,5);
    sample s2(s1);
    return 0;
}
```



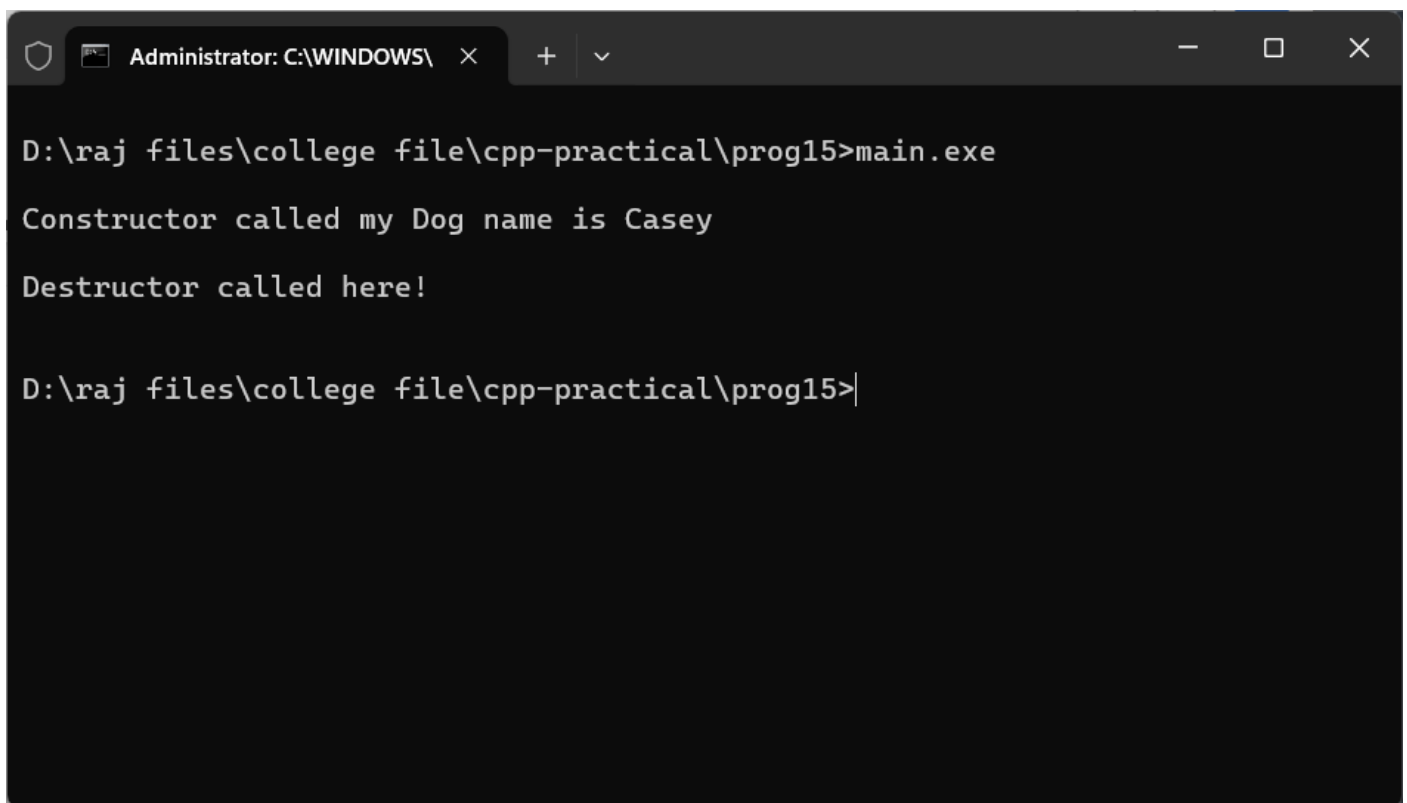
The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following sequence of commands and outputs:

```
D:\raj files\college file\cpp-practical\prog14>main.exe
Sum of two numbers : 5
Sum of numbers : 113
Error! single value supplied
Sum of numbers : 202

D:\raj files\college file\cpp-practical\prog14>|
```

Question 14. Write a program to use the concept of constructor overloading.

```
#include<iostream>
using namespace std;
class addition
{
    public:
        addition(int a){cout << "Error! single value
supplied"<<endl;}
        addition(int a, int b){cout << "Sum of two numbers : "
<< a + b << endl;}
        addition(int a, int b, int c){cout << "Sum of numbers :
" << a+b+c << endl;}
        addition(int a, int b, int c, int d){cout << "Sum of
numbers : " << a+b+c+d << endl;}
};
int main()
{
    addition sum1(2,3), sum2(4, 8, 12, 89), sum3(9), sum4(78,
56, 68);
    return 0;
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following sequence of events:

```
D:\raj files\college file\cpp-practical\prog15>main.exe
Constructor called my Dog name is Casey
Destructor called here!

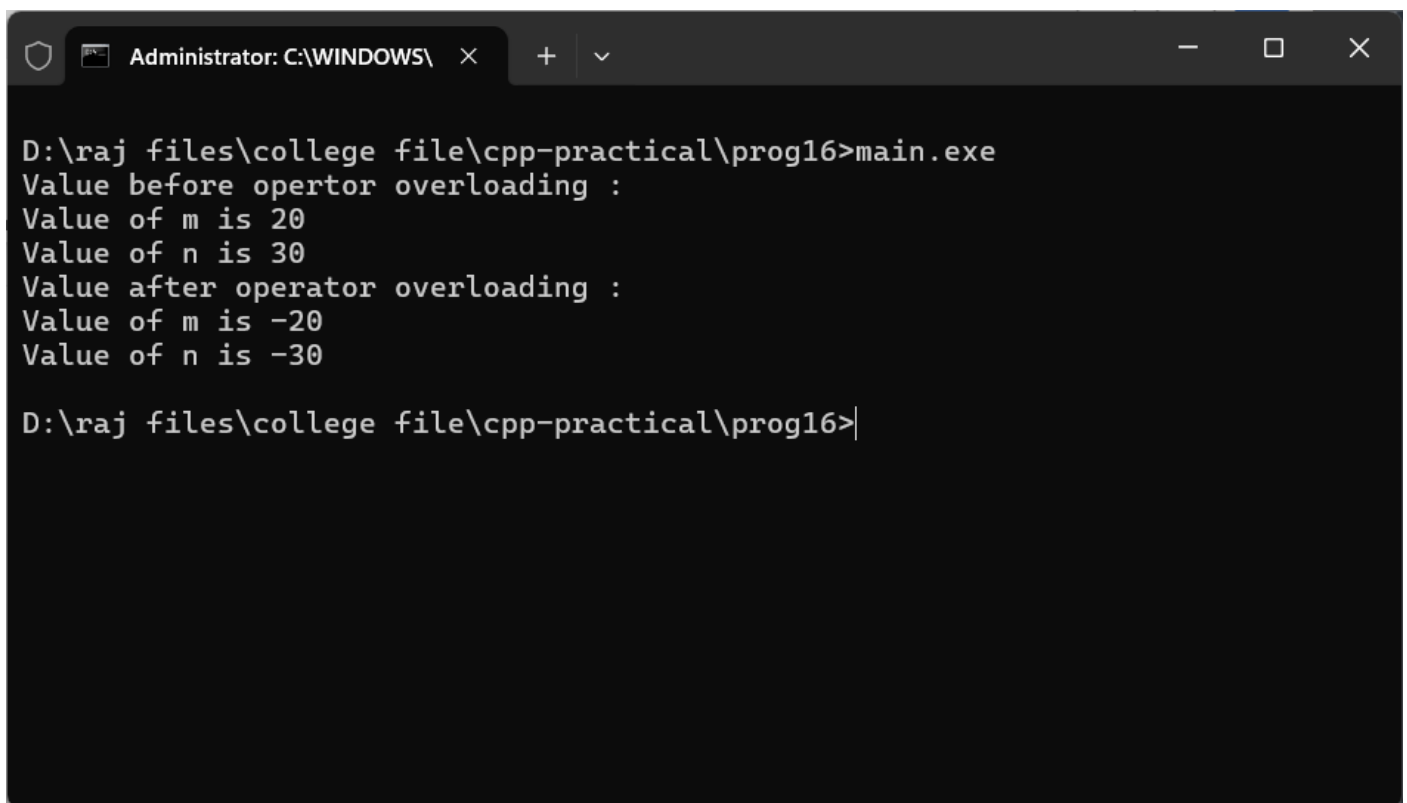
D:\raj files\college file\cpp-practical\prog15>|
```


Question 15. Write a program to use the concept of destructor.

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

class Dog
{
    string name;
public:
    Dog(string dog_name):name(dog_name){cout <<
"\nConstructor called my Dog name is " << name<<endl<<endl;}
    ~Dog(){cout << "Destructor called here!"<<endl<<endl;}
};

int main()
{
    Dog dog1("Casey");
    return 0;
}
```



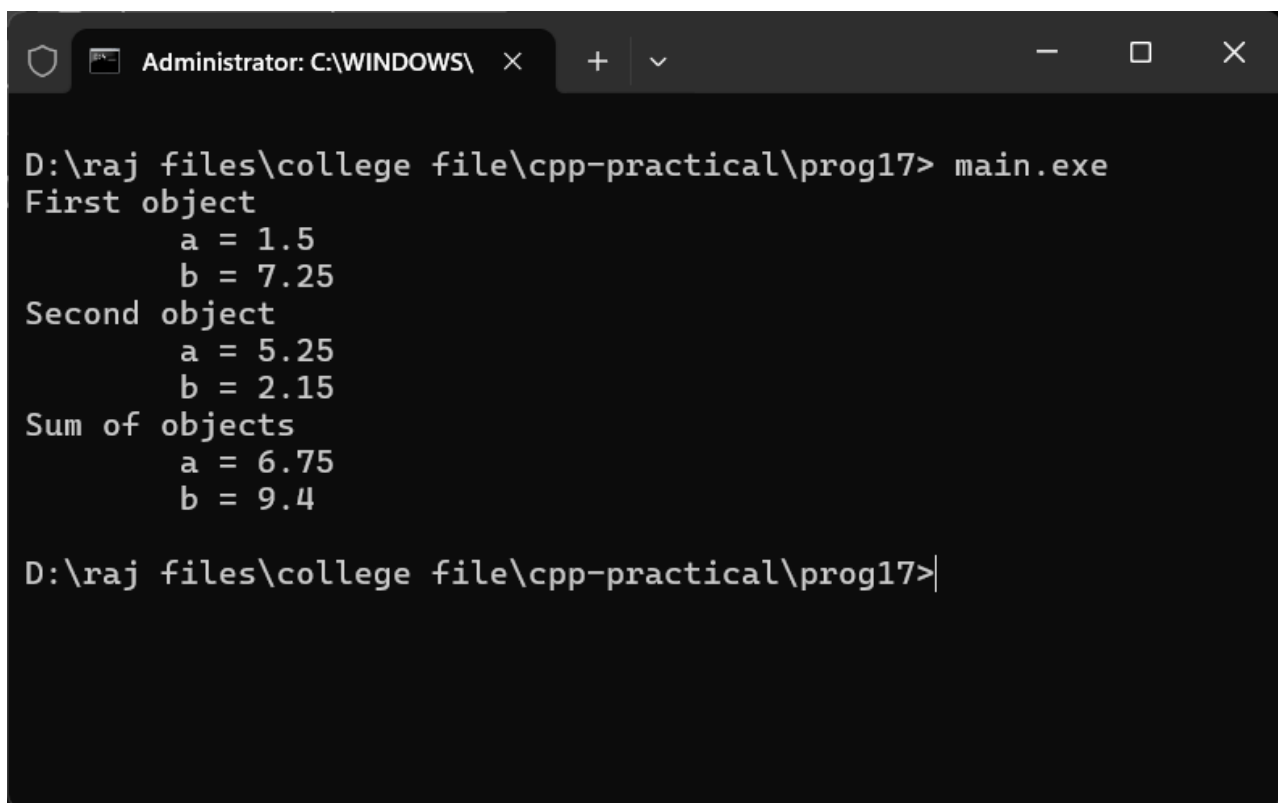
The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following output:

```
D:\raj files\college file\cpp-practical\prog16>main.exe
Value before opertor overloading :
Value of m is 20
Value of n is 30
Value after operator overloading :
Value of m is -20
Value of n is -30

D:\raj files\college file\cpp-practical\prog16>
```

Question 16. Write a program to overload the unary operator by using member function.

```
#include<iostream>
using namespace std;
class sample
{
    int m, n;
public:
    void getdata(int a, int b);
    void display();
    void operator - ();
};
void sample:: getdata(int a, int b){m = a; n = b;}
void sample:: display(){cout << "Value of m is " << m <<
"\nValue of n is " << n << endl;}
void sample:: operator - (){m = -m; n = -n;}
int main()
{
    sample obj;
    obj.getdata(20, 30);
    cout << "Value before opertor overloading :\n";
    obj.display();
    -obj;
    cout << "Value after operator overloading :\n";
    obj.display();
    return 0;
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows the following output:

```
D:\raj files\college file\cpp-practical\prog17> main.exe
First object
    a = 1.5
    b = 7.25
Second object
    a = 5.25
    b = 2.15
Sum of objects
    a = 6.75
    b = 9.4

D:\raj files\college file\cpp-practical\prog17>|
```

Question 17. Write a program to overload the binary operator by using friend function.

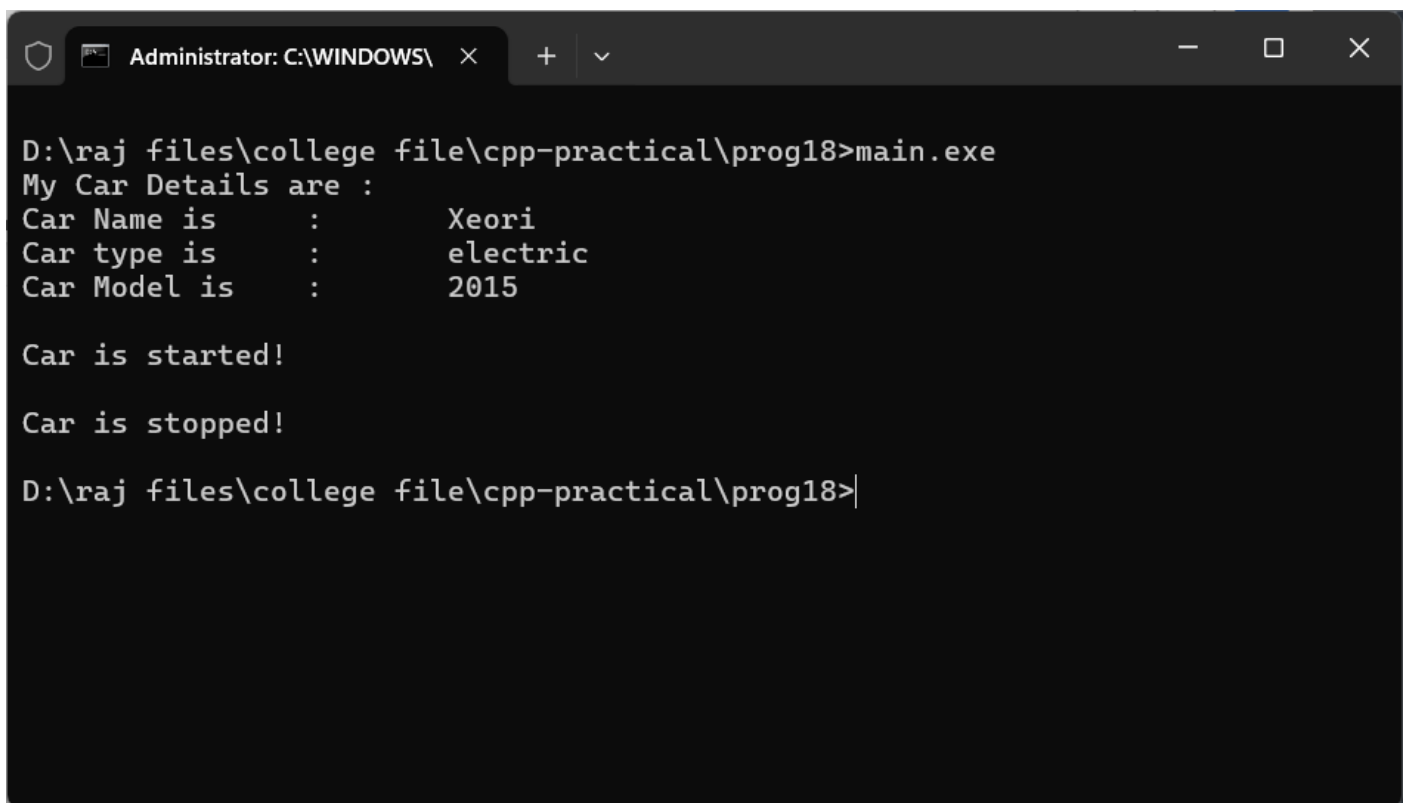
```
#include<iostream>
using namespace std;
class sample
{
    float a,b;
public:
    sample(){};
    sample(float x, float y):a(x), b(y){}
    friend sample operator + (sample, sample);
    void display(){cout<<"\ta = "<<a<<endl;cout<<"\tb = "<<b<<endl;}
};

sample operator + (sample s1, sample s2)
{
    sample temp;
    temp.a = s1.a + s2.a;
    temp.b = s1.b + s2.b;
    return temp;
}

int main()
{
    sample x, y, z;
    x = sample(1.5, 7.25);
    y = sample(5.25, 2.15);
```



```
z = x + y;
    cout << "First object"<<endl;
    x.display();
    cout << "Second object" << endl;
    y.display();
    cout << "Sum of objects" << endl;
    z.display();
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows 'Administrator: C:\WINDOWS\' with standard window controls. The command prompt displays the execution of 'main.exe' from the directory 'D:\raj files\college file\cpp-practical\prog18'. The program outputs car details: 'My Car Details are :', 'Car Name is : Xeori', 'Car type is : electric', and 'Car Model is : 2015'. It then prints 'Car is started!' and 'Car is stopped!' before returning to the command prompt.

```
D:\raj files\college file\cpp-practical\prog18>main.exe
My Car Details are :
Car Name is      :      Xeori
Car type is     :      electric
Car Model is    :      2015

Car is started!

Car is stopped!

D:\raj files\college file\cpp-practical\prog18>
```

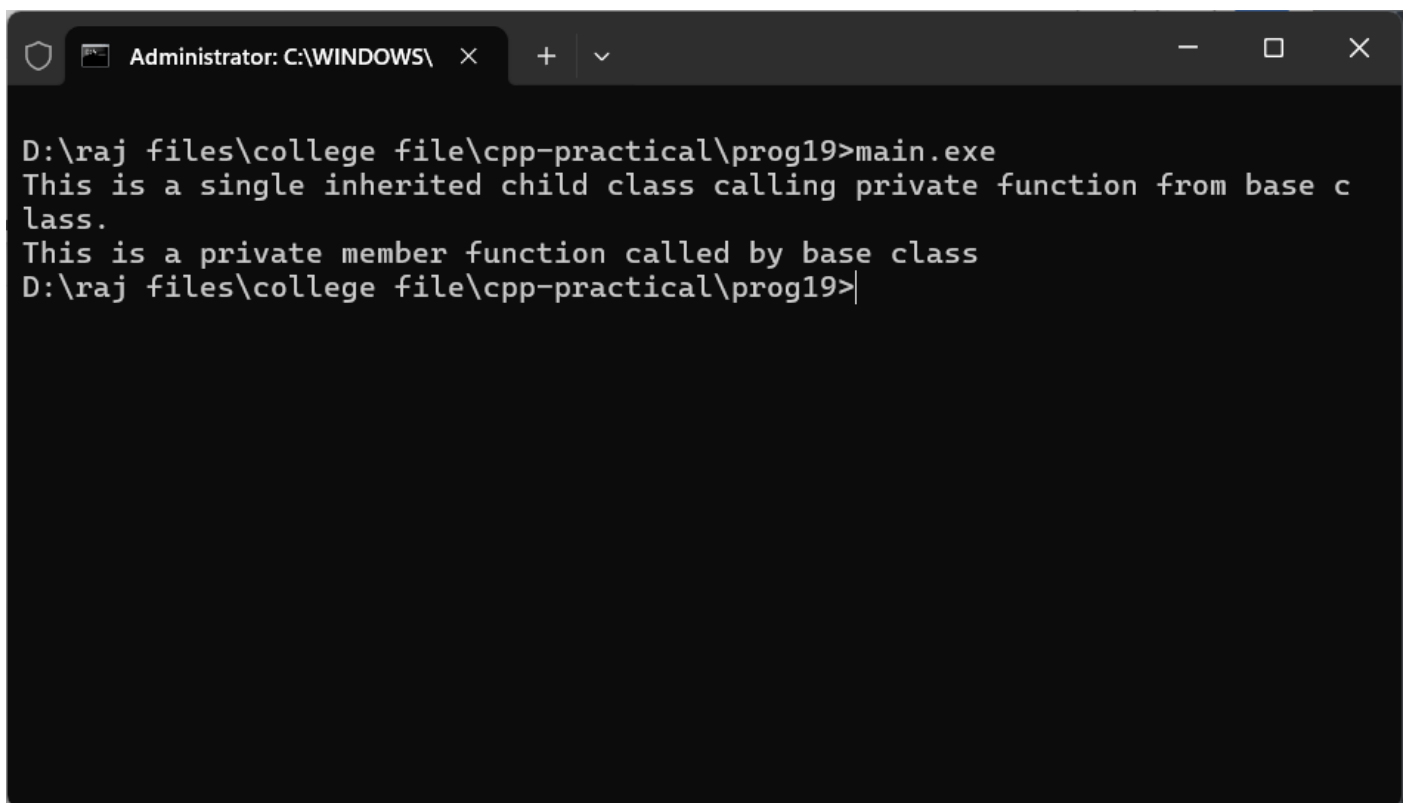

Question 18. Write a program to use the concept of single inheritance (by using public method).

```
#include<iostream>
#include <string>
using namespace std;
class Car
{
    public:
        string name, type;
        int model;
        void car_info()
        {
            cout << "My Car Details are :"<<endl;
            cout << "Car Name is \t:\t" << name << endl;
            cout << "Car type is \t:\t" << type << endl;
            cout << "Car Model is \t:\t" << model <<endl;
        }
        void start(){cout << "\nCar is started!" << endl;}
        void stop(){cout << "\nCar is stopped!" << endl;}
};
class EV_Car : public Car
{
    public:
        EV_Car(string ev_name,int ev_model)
        {
            type = "electric";
            name = ev_name;
```



```
model = ev_model;
        car_info();
    }
};

int main()
{
    EV_Car car1("Xeori", 2015);
    car1.start();
    car1.stop();
    return 0;
}
```

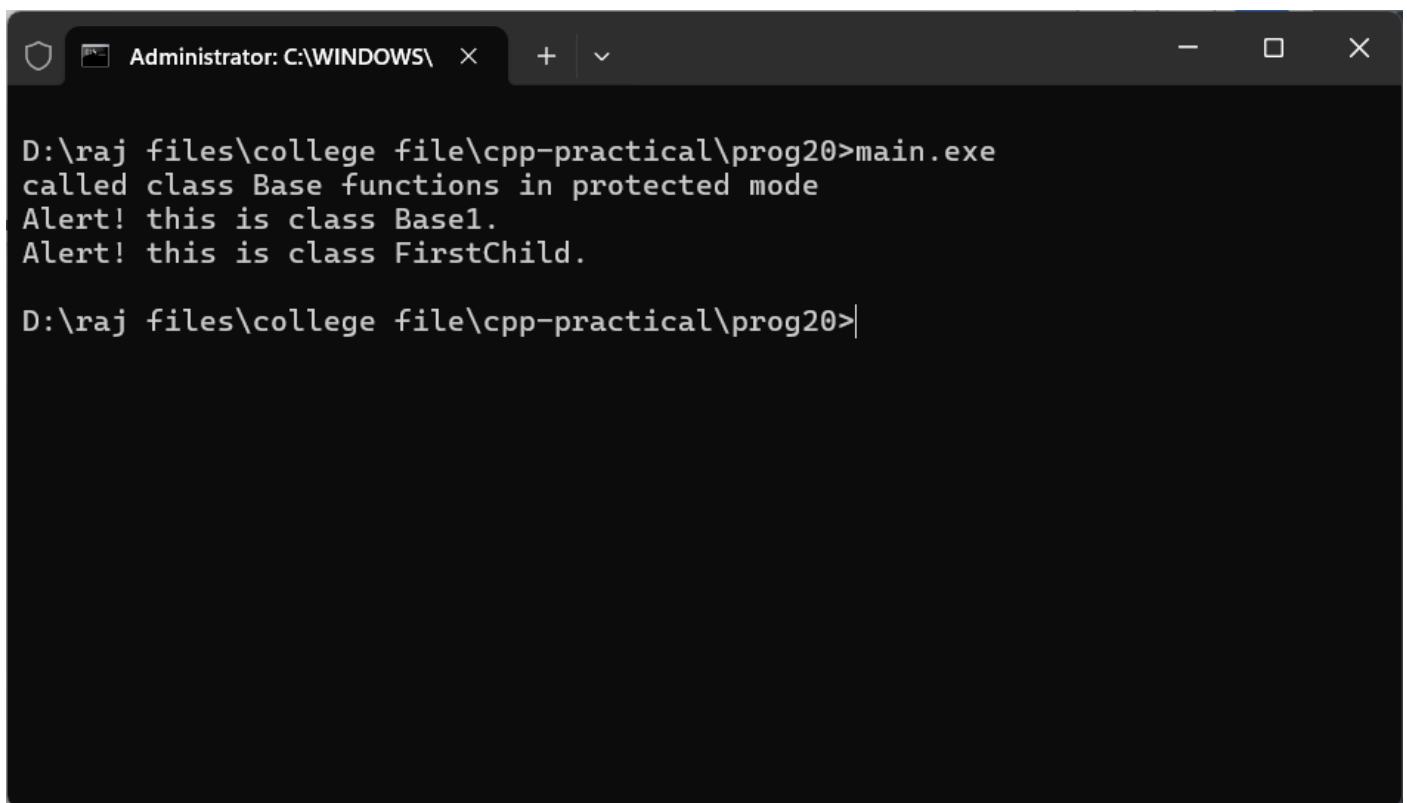


The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The text inside the window shows the execution of a program named "main.exe" from the directory "D:\raj files\college file\cpp-practical\prog19". The output of the program is displayed in the command prompt, showing two lines of text: "This is a single inherited child class calling private function from base class." and "This is a private member function called by base class". The command prompt is currently at the "D:\raj files\college file\cpp-practical\prog19>" prompt.

```
D:\raj files\college file\cpp-practical\prog19>main.exe
This is a single inherited child class calling private function from base c
lass.
This is a private member function called by base class
D:\raj files\college file\cpp-practical\prog19>
```

Question 19. Write a program to use the concept of single inheritance (by using private method).

```
#include <iostream>
using namespace std;
class Base
{
    public:
        void display(){cout<<"This is a private member function
called by base class";}
};
class Child:private Base
{
    public:
        Child()
        {
            cout<<"This is a single inherited child class
calling private function from base class."<<endl;
            display();
        }
};
int main()
{
    Child c1;
    return 0;
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The text inside the window shows the execution of a program named "main.exe" from the directory "D:\raj files\college file\cpp-practical\prog20". The program outputs the following text:

```
D:\raj files\college file\cpp-practical\prog20>main.exe
called class Base functions in protected mode
Alert! this is class Base1.
Alert! this is class FirstChild.

D:\raj files\college file\cpp-practical\prog20>|
```

Question 20. Write a program to sue the concept of multilevel inheritance (by using protected method).

```
#include<iostream>

using namespace std;

class Base
{
    protected:
        void display1(){cout<<"Alert! this is class
Base1."<<endl;}
};

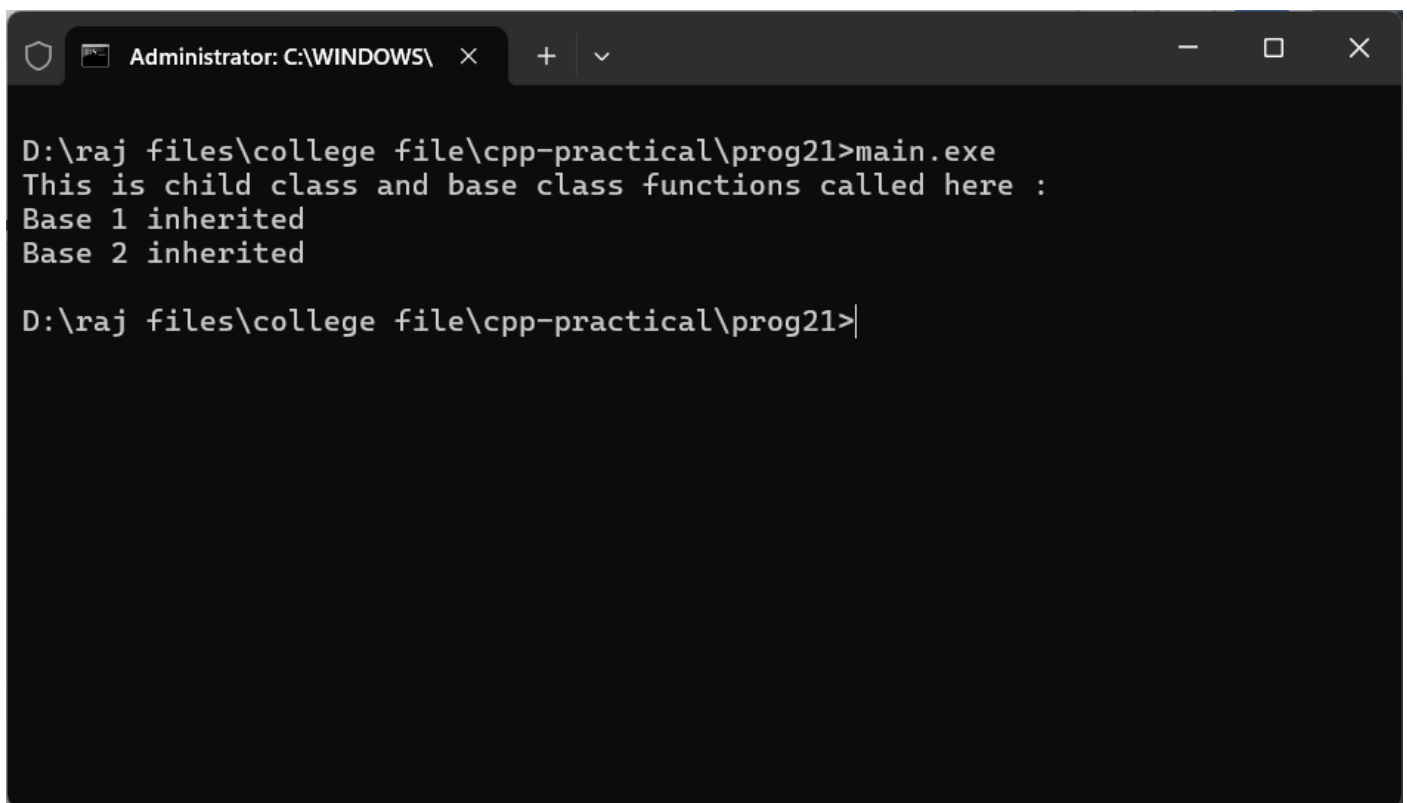
class FirstChild: protected Base
{
    protected:
        void display2(){cout<<"Alert! this is class
FirstChild."<<endl;}
};

class SecondChild: public FirstChild
{
    public:
        SecondChild()
        {
            cout<<"called class Base functions in protected
mode"<<endl;
            display1();
            display2();
        }
};

int main()
```



```
{  
    SecondChild c1;  
    return 0;  
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following text:

```
D:\raj files\college file\cpp-practical\prog21>main.exe
This is child class and base class functions called here :
Base 1 inherited
Base 2 inherited

D:\raj files\college file\cpp-practical\prog21>|
```

Question 21. Write a program by using concept of multiple inheritance.

```
#include<iostream>

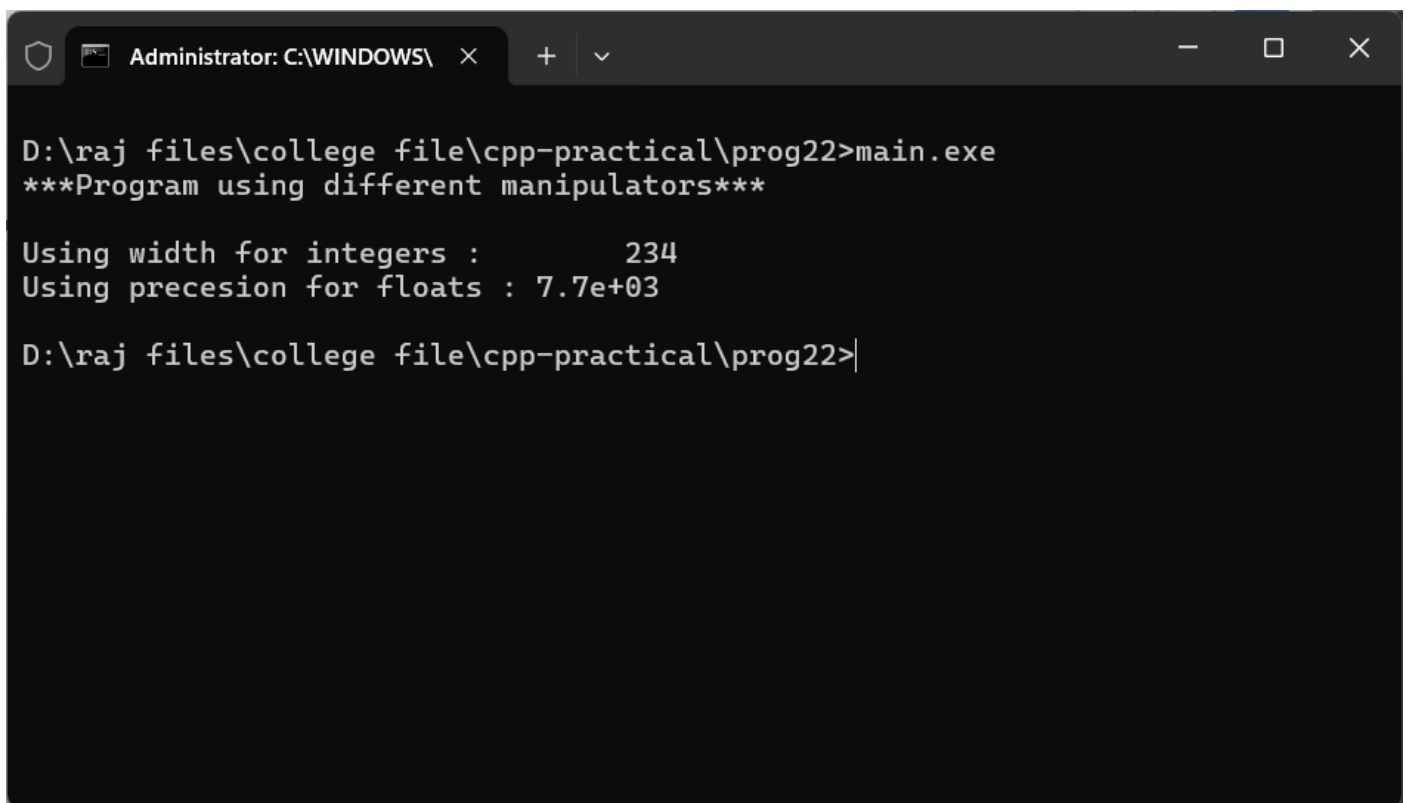
using namespace std;

class Base1{public:void fun1(){cout<<"Base 1
inherited"<<endl;}};

class Base2{public:void fun2(){cout<<"Base 2
inherited"<<endl;}};

class Child:public Base1, public Base2
{
    public:
        Child()
        {
            cout<<"This is child class and base class functions
called here :"<<endl;
            fun1();
            fun2();
        }
};

int main()
{
    Child c1;
    return 0;
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The command prompt shows the following text:

```
D:\raj files\college file\cpp-practical\prog22>main.exe
***Program using different manipulators***

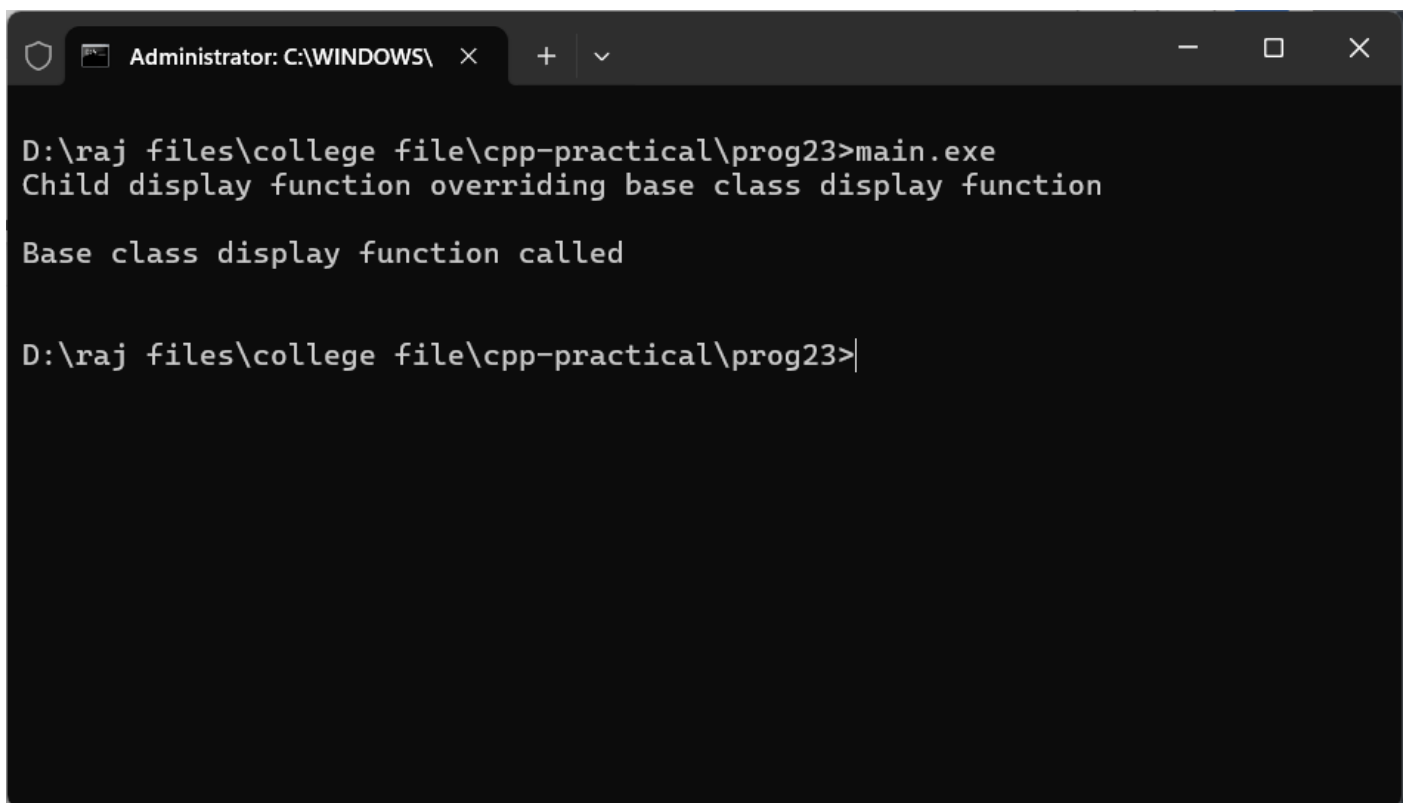
Using width for integers :      234
Using precesion for floats : 7.7e+03

D:\raj files\college file\cpp-practical\prog22>|
```

The text is displayed in a monospaced font. The output shows the results of the program, which uses different manipulators for formatting integers and floats. The integer 234 is formatted with a width of 234, and the float 7.7e+03 is formatted with a precision of 7.7e+03.

Question 22. Write a program by using various Manipulators.

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    cout<< "***Program using different manipulators***\n";
    cout<< "\nUsing width for integers : " << setw(10) << 234
<<endl;
    cout<< "Using precesion for floats : " << setprecision(2)
<< 7678.328403 <<endl;
    return 0;
}
```



The image shows a Windows command prompt window titled "Administrator: C:\WINDOWS\". The window has a dark background and a light-colored text. The text inside the window is as follows:

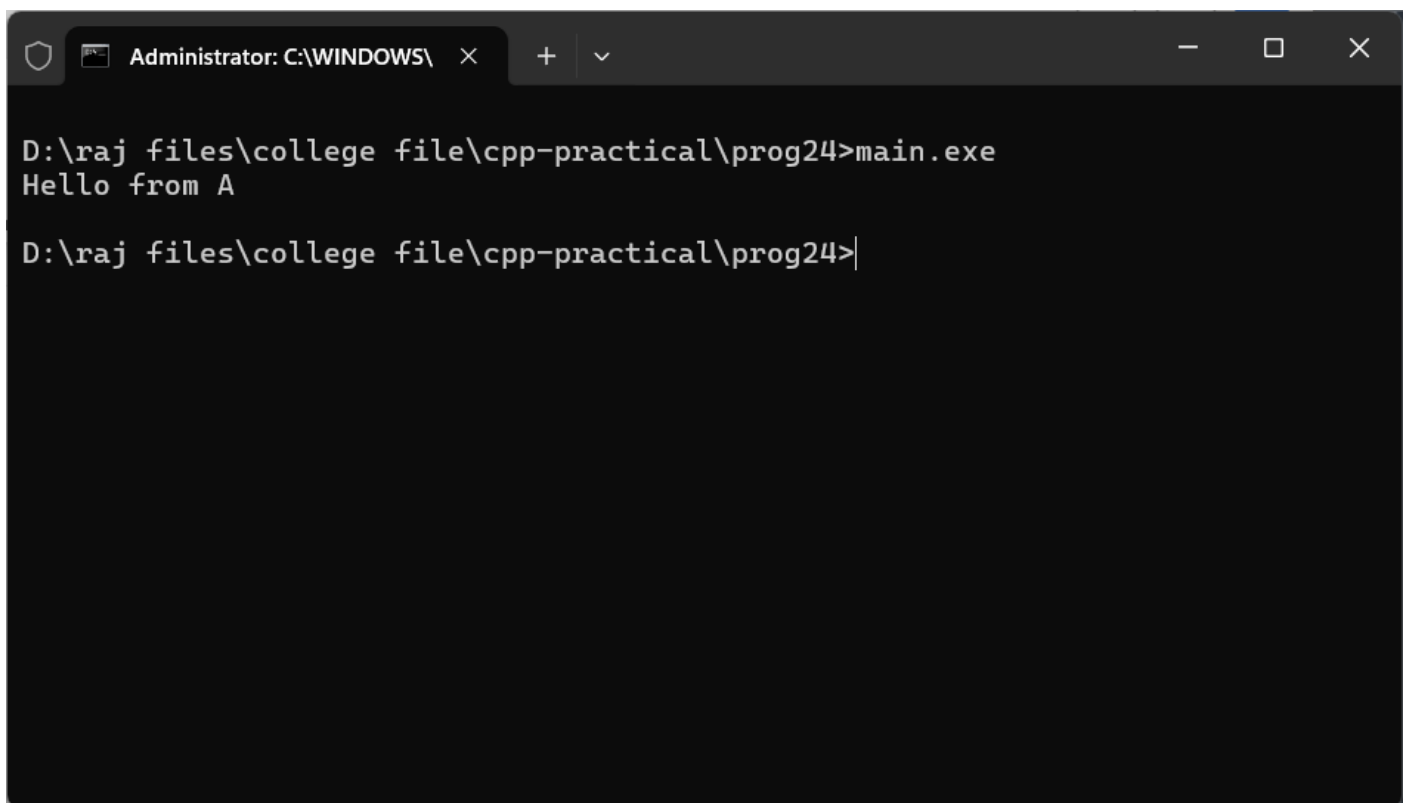
```
D:\raj files\college file\cpp-practical\prog23>main.exe
Child display function overriding base class display function

Base class display function called

D:\raj files\college file\cpp-practical\prog23>|
```

Question 23. Write a program to use the concept of function overriding.

```
#include<iostream>
using namespace std;
class Base
{
    public:
        void display(){cout << "Base class display function
called\n"<< endl;}
};
class Child : public Base
{
    public:
        void display(){cout<<"Child display function overriding
base class display function\n"<<endl;}
};
int main()
{
    Child c1;
    c1.display();
    c1.Base :: display();
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by a close button (X). To the right of the title bar are standard window controls: a plus sign, a downward arrow, a minus sign, a maximize button (square), and a close button (X). The command prompt area has a black background with white text. The first line shows the command `D:\raj files\college file\cpp-practical\prog24>main.exe`. The second line shows the output `Hello from A`. The third line shows the prompt `D:\raj files\college file\cpp-practical\prog24>` with a cursor at the end.

```
Administrator: C:\WINDOWS\ X + v - □ X  
D:\raj files\college file\cpp-practical\prog24>main.exe  
Hello from A  
D:\raj files\college file\cpp-practical\prog24>
```


Question 24. Write a program to use the concept of virtual base class.

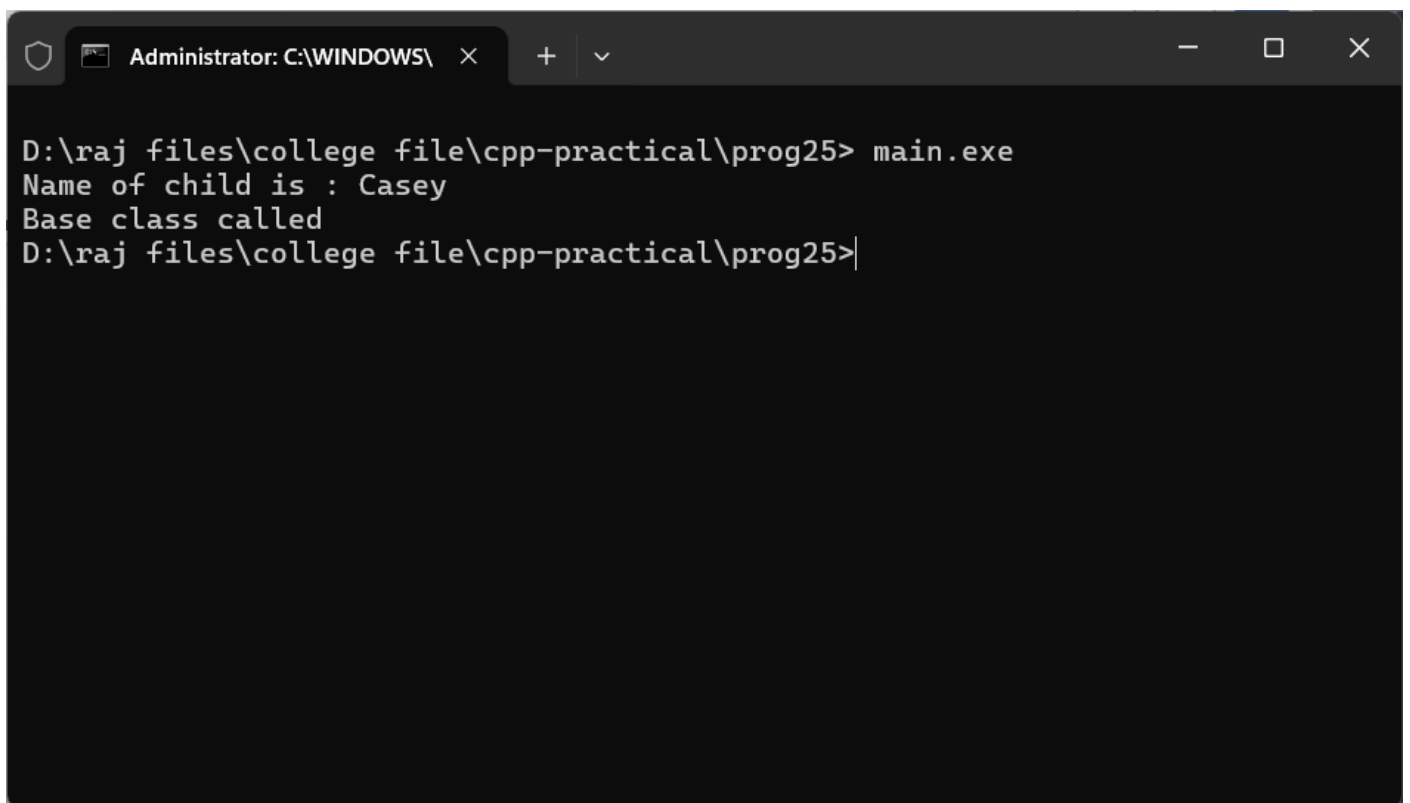
```
#include<iostream>

using namespace std;

class A
{
    public: void show(){cout <<"Hello from A \n";}
};

class B : public virtual A {};
class C : public virtual A {};
class D : public B, public C {};

int main()
{
    D object;
    object.show();
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt itself has a black background with white text. The first line shows the command "D:\raj files\college file\cpp-practical\prog25> main.exe". The next two lines show the program's output: "Name of child is : Casey" and "Base class called". The final line shows the command prompt ready for input: "D:\raj files\college file\cpp-practical\prog25>".

```
Administrator: C:\WINDOWS\ × + ▾  
D:\raj files\college file\cpp-practical\prog25> main.exe  
Name of child is : Casey  
Base class called  
D:\raj files\college file\cpp-practical\prog25>
```

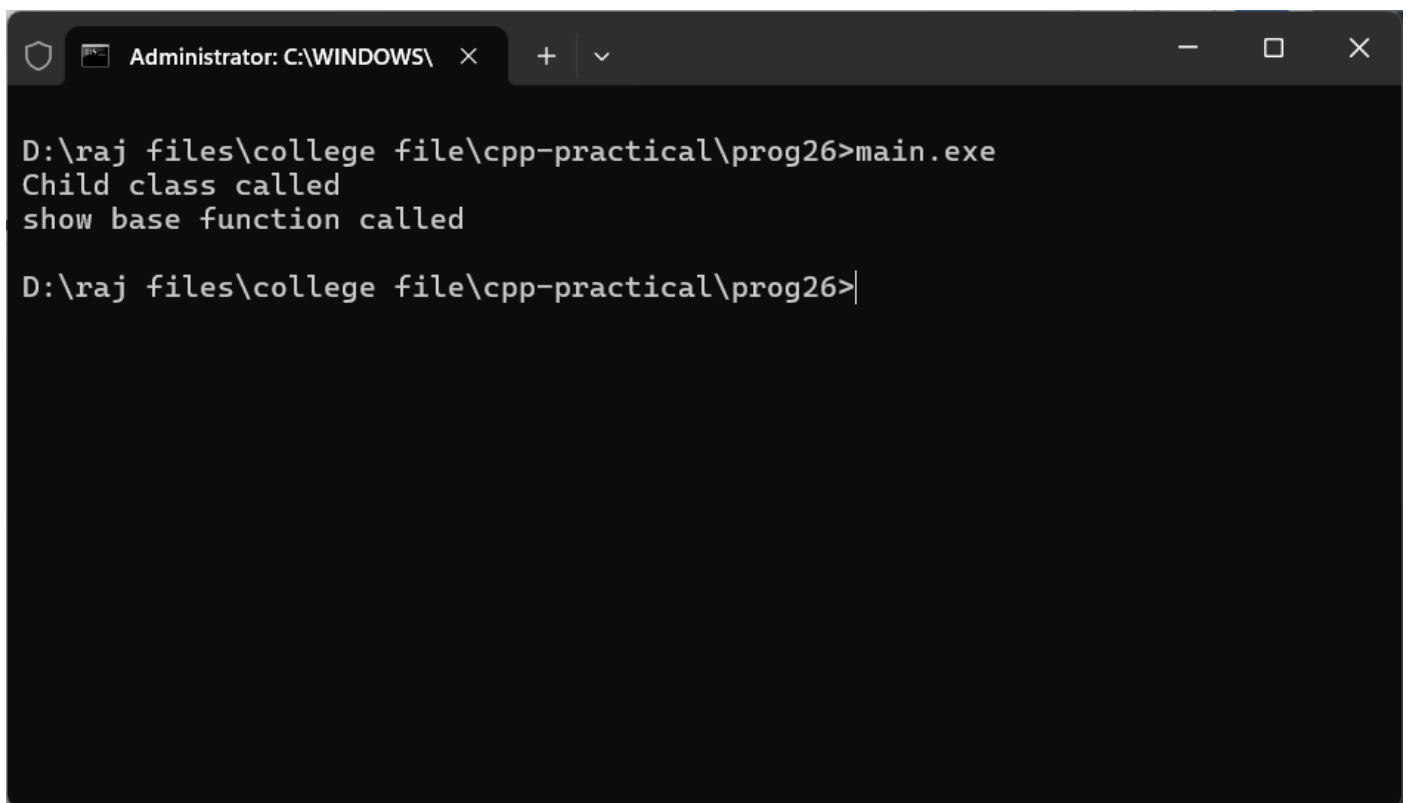
Question 25. Write a program to use the concept of inheritance and parameterized constructor.

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

class Base{public : void display(){cout <<"Base class
called";}};

class Child : public Base
{
    string name;
public:
    Child(string child_name)
    {
        name = child_name;
        cout<<"Name of child is : "<<name<<endl;
        display();
    }
};

int main()
{
    Child c1("Casey");
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar at the top shows a shield icon, a small application icon, and the text "Administrator: C:\WINDOWS\" followed by window control buttons (close, maximize, minimize). The command prompt itself has a black background with white text. The text shows the current directory as "D:\raj files\college file\cpp-practical\prog26" and the execution of "main.exe". The output of the program is "Child class called" followed by "show base function called" on a new line. The prompt is ready for the next command.

```
D:\raj files\college file\cpp-practical\prog26>main.exe
Child class called
show base function called
D:\raj files\college file\cpp-practical\prog26>|
```

Question 26. Write a program to use the concept of virtual function.

```
#include<iostream>
using namespace std;
class Base
{
public:
    virtual void display(){cout << "Base class function"
<<endl;}
    void show(){cout << "show base function called"<<endl;}
};
class Child: public Base
{
public:
    void display(){cout <<"Child class called"<<endl;}
    void show(){cout << "child show function
called"<<endl;}
};
int main()
{
    Base * ptr;
    Child c1;
    ptr = &c1;
    ptr -> display();
    ptr -> show();
    return 0;}
```