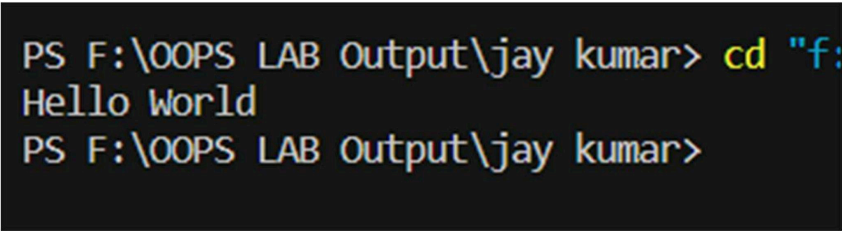


PRACTICAL NO. 1

Write a Program to Print Hello World .

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Hello World";
    return 0;
}
```

OUTPUT



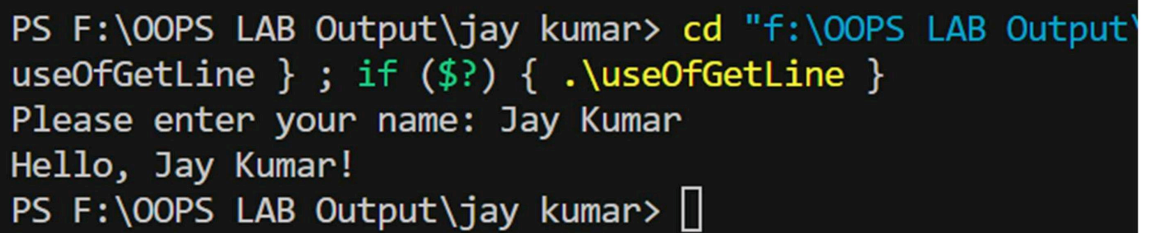
```
PS F:\OOPS LAB Output\jay kumar> cd "f:
Hello World
PS F:\OOPS LAB Output\jay kumar>
```

PRACTICAL NO. 2

Write a Program to Print Your Name.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string name;
    cout << "Please enter your name: ";
    getline(cin, name);
    cout << "Hello, " << name << "!" << endl;
    return 0;
}
```

OUTPUT



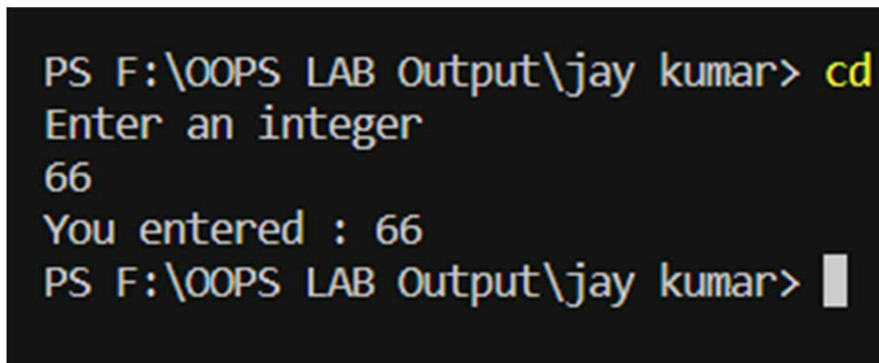
```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB Output\
useOfGetLine } ; if ($?) { .\useOfGetLine }
Please enter your name: Jay Kumar
Hello, Jay Kumar!
PS F:\OOPS LAB Output\jay kumar> □
```

PRACTICAL NO. 3

Write a Program to Print an Integer entered by the User at run time.

```
#include<iostream>
using namespace std;
int main()
{
    int num;
    cout<<"Enter an integer"<<endl;
    cin>>num;
    cout<<"You entered : "<<num;
    return 0;
}
```

OUTPUT



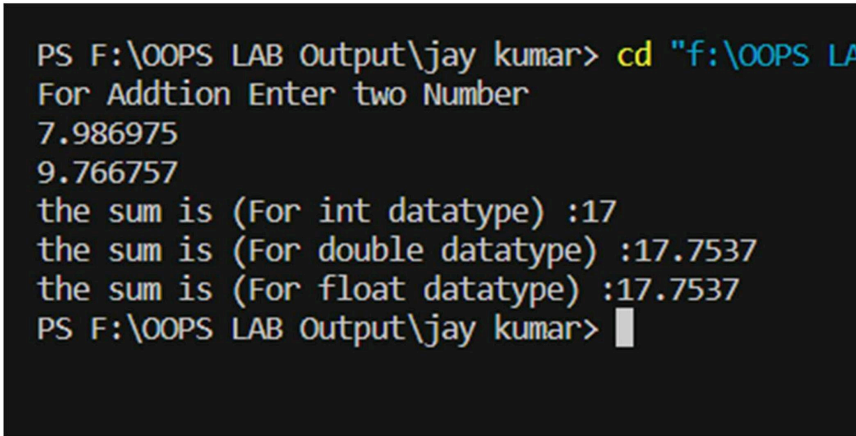
```
PS F:\OOPS LAB Output\jay kumar> cd
Enter an integer
66
You entered : 66
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 4

Write a Program to add two numbers having int , double,float data type.

```
#include<iostream>
using namespace std;
int main()
{
    double num1;
    double num2;
    cout<<"For Addtion Enter two Number"<<endl;
    cin>>num1>>num2;
    int sum1 = num1+ num2;
    double sum2 = num1+ num2;
    float sum3 = num1+ num2;
    cout<<"the sum is (For int datatype) :"<<sum1<<endl;
    cout<<"the sum is (For double datatype) :"<<sum2<<endl;
    cout<<"the sum is (For float datatype) :"<<sum3<<endl;
    return 0;
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LA
For Addtion Enter two Number
7.986975
9.766757
the sum is (For int datatype) :17
the sum is (For double datatype) :17.7537
the sum is (For float datatype) :17.7537
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 5

Write a Program to demonstrate unary operator in C++.

```
#include <iostream>
using namespace std;
int main(){
    int num = 10;
    cout << "Initial value of num: " << num << endl;
    // Unary increment operator (++)
    cout << "After pre-increment (++num): " << ++num << endl;
    // Unary decrement operator (--)
    cout << "After pre-decrement (--num): " << --num << endl;
    // Unary plus operator (+)
    cout << "Unary plus operator (+num): " << +num << endl;
    // Unary minus operator (-)
    cout << "Unary minus operator (-num): " << -num << endl;
    // Unary logical NOT operator (!)
    bool isTrue = true;
    cout << "Unary logical NOT operator (!isTrue): " << !isTrue << endl;
    // Unary post-increment operator (num++)
    cout << "After post-increment (num++): " << num++ << endl;
    cout << "Value of num after post-increment: " << num << endl;
    return 0;
}
```

OUTPUT

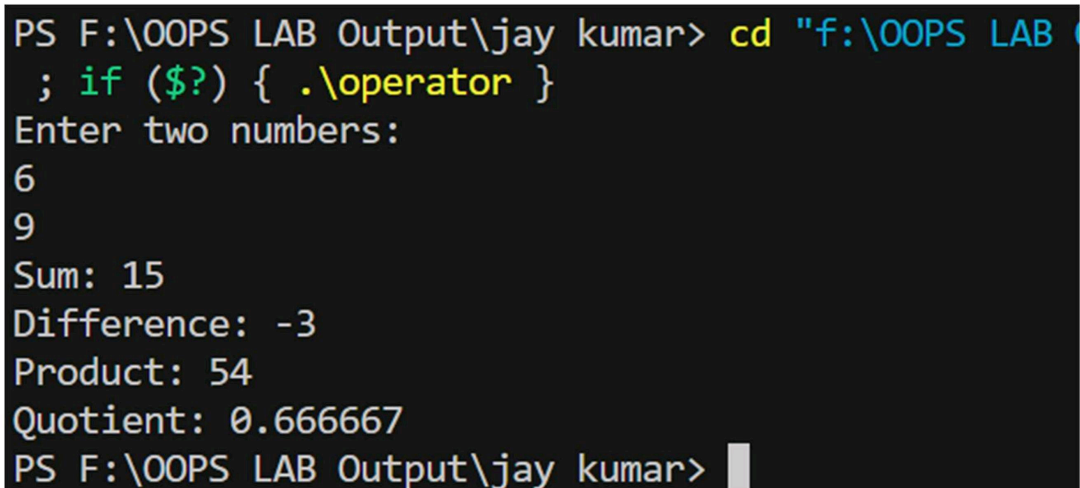
```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB Output\jay kumar" & g++ incrementdecrement.cpp & if ($?) { .\incrementdecrement.exe }
Initial value of num: 10
After pre-increment (++num): 11
After pre-decrement (--num): 10
Unary plus operator (+num): 10
Unary minus operator (-num): -10
Unary logical NOT operator (!isTrue): 0
After post-increment (num++): 10
Value of num after post-increment: 11
```

PRACTICAL NO. 6

Write a Program to add, subtract, multiply, divide Two numbers.

```
#include <iostream>
using namespace std;
int main() {
    double num1, num2;
    cout << "Enter two numbers: "<<endl;
    cin >> num1 >> num2;
    cout << "Sum: " << num1 + num2 << endl;
    cout << "Difference: " << num1 - num2 << endl;
    cout << "Product: " << num1 * num2 << endl;
    if (num2 != 0)
        cout << "Quotient: " << num1 / num2 << endl;
    else
        cout << "Division by zero is not allowed." << endl;
    return 0;
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB ...
; if ($?) { .\operator }
Enter two numbers:
6
9
Sum: 15
Difference: -3
Product: 54
Quotient: 0.666667
PS F:\OOPS LAB Output\jay kumar>
```

PRACTICAL NO. 7

Write a Program to check whether the given no. is odd or even.

```
#include <iostream>
using namespace std;
int main() {
    int number;
    cout << "Enter an integer: ";
    cin >> number;

    if (number % 2 == 0) {
        cout << number << " is even." << endl;
    } else {
        cout << number << " is odd." << endl;
    }
    return 0;
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "
Enter an integer: 78780
78780 is even.
PS F:\OOPS LAB Output\jay kumar> █
```

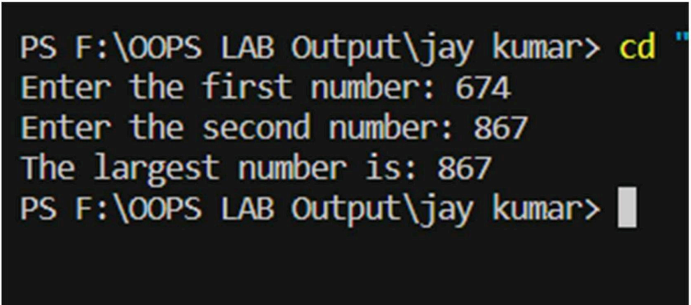
```
PS F:\OOPS LAB Output\jay kumar> cd "
Enter an integer: 787
787 is odd.
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 8

Write a Program to find the Largest of Two Number.

```
#include <iostream>
using namespace std;
int main()
{
    double num1, num2;
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    if (num1 > num2) {
        cout << "The largest number is: " << num1 << endl;
    } else if (num2 > num1) {
        cout << "The largest number is: " << num2 << endl;
    } else {
        cout << "Both numbers are equal." << endl;
    }
    return 0;
}
```

OUTPUT



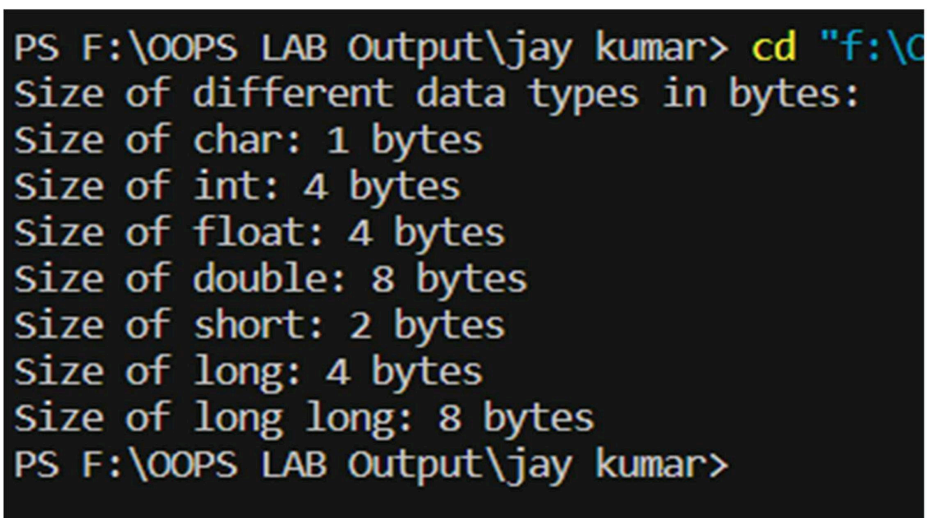
```
PS F:\OOPS LAB Output\jay kumar> cd "
Enter the first number: 674
Enter the second number: 867
The largest number is: 867
PS F:\OOPS LAB Output\jay kumar> █
```


PRACTICAL NO. 9

Write a Program to demonstrate the use of sizeof operator to determine the size of different datatype.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Size of different data types in bytes:" << endl;
    cout << "Size of char: " << sizeof(char) << " bytes" << endl;
    cout << "Size of int: " << sizeof(int) << " bytes" << endl;
    cout << "Size of float: " << sizeof(float) << " bytes" << endl;
    cout << "Size of double: " << sizeof(double) << " bytes" << endl;
    cout << "Size of short: " << sizeof(short) << " bytes" << endl;
    cout << "Size of long: " << sizeof(long) << " bytes" << endl;
    cout << "Size of long long: " << sizeof(long long) << " bytes" << endl;
    return 0;
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\c
Size of different data types in bytes:
Size of char: 1 bytes
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of short: 2 bytes
Size of long: 4 bytes
Size of long long: 8 bytes
PS F:\OOPS LAB Output\jay kumar>
```

PRACTICAL NO. 10

Write a Program to that check if year is leap or not using if else.

```
#include <iostream>
using namespace std;
int main() {
    int year;
    cout << "Enter a year: ";
    cin >> year;

    if (year % 4 == 0) {
        cout << year << " is a leap year." << endl;
    } else {
        cout << year << " is not a leap year." << endl;
    }
    return 0;
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "f
Enter a year: 2024
2024 is a leap year.
PS F:\OOPS LAB Output\jay kumar> █
```

```
PS F:\OOPS LAB Output\jay kumar> cd "
Enter a year: 2023
2023 is not a leap year.
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 11

Write a Program using switch case to display corresponding day on numeric input(1 to 7).

```
#include <iostream>

using namespace std;

int main() {
    int dayNumber;

    cout << "Enter a number (1-7) to display the corresponding day of the
    week: ";

    cin >> dayNumber;

    switch (dayNumber) {
        case 1:
            cout << "Sunday" << endl;
            break;
        case 2:
            cout << "Monday" << endl;
            break;
        case 3:
            cout << "Tuesday" << endl;
            break;
        case 4:
            cout << "Wednesday" << endl;
            break;
        case 5:
            cout << "Thursday" << endl;
            break;
        case 6:
```

```
cout << "Friday" << endl;
break;
case 7:
cout << "Saturday" << endl;
break;
default:
cout << "Invalid input! Please enter a number between 1 and 7." <<
endl;
break;
}
return 0;
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB Output\jay kumar\" ;
Enter a number (1-7) to display the corresponding day of the week: 6
Friday
PS F:\OOPS LAB Output\jay kumar> █
```

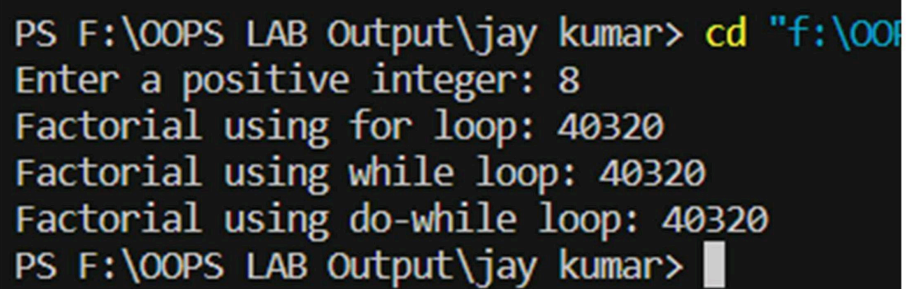
PRACTICAL NO. 12

Write a Program using for while and do while to find factorial of a number.

```
#include <iostream>
using namespace std;
int main() {
    int number;
    cout << "Enter a positive integer: ";
    cin >> number;
    int factorial_for = 1;
    for (int i = 1; i <= number; i++) {
        factorial_for *= i;
    }
    cout << "Factorial using for loop: " << factorial_for << endl;
    int factorial_while = 1;
    int i = 1;
    while (i <= number) {
        factorial_while *= i;
        i++;
    }
    cout << "Factorial using while loop: " << factorial_while << endl;
    int factorial_do_while = 1;
    int j = 1;
    do {
        factorial_do_while *= j;
        j++;
    }
```

```
    } while (j <= number);  
  
    cout << "Factorial using do-while loop: " << factorial_do_while <<  
endl;  
  
    return 0;  
}
```

OUTPUT



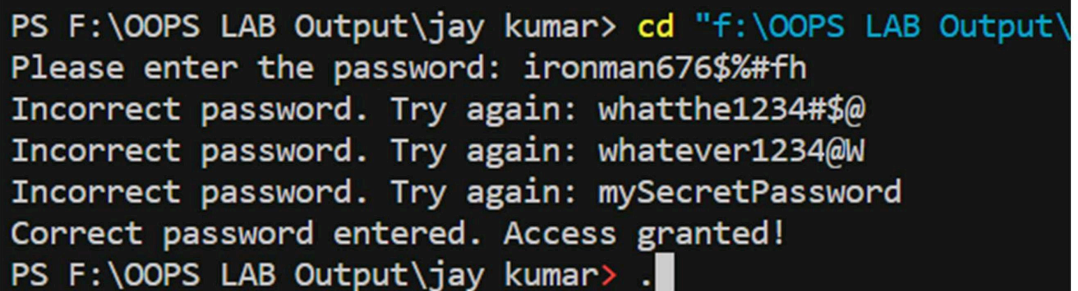
```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB Output\jay kumar"  
Enter a positive integer: 8  
Factorial using for loop: 40320  
Factorial using while loop: 40320  
Factorial using do-while loop: 40320  
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 13

Write a program that prompts the user for a password until they enter the correct password using while loop.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    const string correctPassword = "mySecretPassword";
    string enteredPassword;
    cout << "Please enter the password: ";
    cin >> enteredPassword;
    while (enteredPassword != correctPassword) {
        cout << "Incorrect password. Try again: ";
        cin >> enteredPassword;
    }
    cout << "Correct password entered. Access granted!" << endl;
    return 0;
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS LAB Output\"
Please enter the password: ironman676$%#fh
Incorrect password. Try again: whatthe1234#$$@
Incorrect password. Try again: whatever1234@W
Incorrect password. Try again: mySecretPassword
Correct password entered. Access granted!
PS F:\OOPS LAB Output\jay kumar> .
```

PRACTICAL NO. 14

Write a menu driven program to add,subtract,multiply,divide two number.

```
#include <iostream>
using namespace std;
int main() {
    char choice;
    double num1, num2;
    while (true) {
        cout << "Menu:\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5.
        Quit" << endl;
        cout << "Enter your choice (1/2/3/4/5): ";
        cin >> choice;
        if (choice == '5') {
            cout << "Exiting the program." << endl;
            break;
        }
        cout << "Enter two numbers: ";
        cin >> num1 >> num2;
        switch (choice) {
            case '1':
                cout << "Result: " << num1 + num2 << endl;
                break;
            case '2':
                cout << "Result: " << num1 - num2 << endl;
                break;
```



```

    case '3':
        cout << "Result: " << num1 * num2 << endl;
        break;
    case '4':
        if (num2 != 0)
            cout << "Result: " << num1 / num2 << endl;
        else
            cout << "Cannot divide by zero." << endl;
        break;
    default:
        cout << "Invalid choice. Please select a valid option." << endl;
}
}
return 0;
}

```

OUTPUT

```

PS F:\OOPS LAB Output\jay kumar> cd "f:\OO
}
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Quit
Enter your choice (1/2/3/4/5): 4
Enter two numbers: 7 8
Result: 0.875
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Quit
Enter your choice (1/2/3/4/5): 2
Enter two numbers: 8 4
Result: 4
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Quit
Enter your choice (1/2/3/4/5): 5
Exiting the program.
PS F:\OOPS LAB Output\jay kumar> █

```

PACTICAL NO. 15

Write a program in c++ to demonstrate the use of class,object,reference variables(Bank a/c system).

```
#include <iostream>
#include <string>
using namespace std;
class BankAccount {
public:
    string accountHolder;
    double balance;
    void init(const string& holder, double initialBalance) {
        accountHolder = holder;
        balance = initialBalance;
    }
    void deposit(double amount) {
        balance += amount;
    }
    bool withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            return true;
        }
        return false;
    }
};
int main() {
```

```

BankAccount account;
BankAccount& accountRef = account;
string holder;
double initialBalance, depositAmount, withdrawAmount;
cout << "Enter account holder's name: ";
cin >> holder;
cout << "Enter initial balance: $";
cin >> initialBalance;
accountRef.init(holder, initialBalance);
cout << "Enter the deposit amount: $";
cin >> depositAmount;
accountRef.deposit(depositAmount);
cout << "Enter the withdrawal amount: $";
cin >> withdrawAmount;
if (accountRef.withdraw(withdrawAmount)) {
    cout << "Withdrawal successful." << endl;
} else {
    cout << "Insufficient funds. Withdrawal not allowed." << endl;
}
cout << "Account holder: " << accountRef.accountHolder << ", Balance: $" <<
accountRef.balance << endl;
return 0;
}

```

OUTPUT

```

Enter account holder's name: jay
Enter initial balance: $77678
Enter the deposit amount: $6656
Enter the withdrawal amount: $800
Withdrawal successful.
Account holder: jay, Balance: $83534
PS D:\My coding\physicswallah C++ with DSA
coding files\Extra coding> S

```

PRACTICAL NO. 16

Write a program in c++ to convert the temperature from Fahrenheit to celcius & Celsius to Fahrenheit.

```
#include <iostream>
using namespace std;
double celsiusToFahrenheit(double celsius) {
    return (celsius * 9.0 / 5.0) + 32;
}
double fahrenheitToCelsius(double fahrenheit) {
    return (fahrenheit - 32) * 5.0 / 9.0;
}
int main() {
    int choice;
    double temperature, result;
    do {
        cout << "Menu:" << endl;
        cout << "1. Convert Celsius to Fahrenheit" << endl;
        cout << "2. Convert Fahrenheit to Celsius" << endl;
        cout << "3. Quit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                cout << "Enter temperature in Celsius: ";
                cin >> temperature;
                result = celsiusToFahrenheit(temperature);
```

```

        cout << "Temperature in Fahrenheit: " << result << "F" << endl;
        break;
    case 2:
        cout << "Enter temperature in Fahrenheit: ";
        cin >> temperature;
        result = fahrenheitToCelsius(temperature);
        cout << "Temperature in Celsius: " << result << "C" << endl;
        break;
    case 3:
        cout << "Goodbye!" << endl;
        break;
    default:
        cout << "Invalid choice. Please enter a valid option." << endl;
        break;
    }
} while (choice != 3);
return 0;
}

```

OUTPUT

```

Menu:
1. Convert Celsius to Fahrenheit
2. Convert Fahrenheit to Celsius
3. Quit
Enter your choice: 1
Enter temperature in Celsius: 0
Temperature in Fahrenheit: 32F
Menu:
1. Convert Celsius to Fahrenheit
2. Convert Fahrenheit to Celsius
3. Quit
Enter your choice: 2
Enter temperature in Fahrenheit: 98.6
Temperature in Celsius: 37C
Menu:
1. Convert Celsius to Fahrenheit
2. Convert Fahrenheit to Celsius
3. Quit
Enter your choice: 3
Goodbye!

```

PRACTICAL NO. 17

Write a program in c++ using constructor on Car Dealership.

```
#include <iostream>

#include <string>

using namespace std;

class Car {

public:

    string make;

    string model;

    int year;

    double price;

    // Constructor to initialize car details

    Car(string make, string model, int year, double price) : make(make),
model(model), year(year), price(price) {

    }

    // Display car information

    void display() {

        cout << "Make: " << make << endl;

        cout << "Model: " << model << endl;

        cout << "Year: " << year << endl;

        cout << "Price: $" << price << endl;

    }

};

int main() {

    string make, model;

    int year;
```

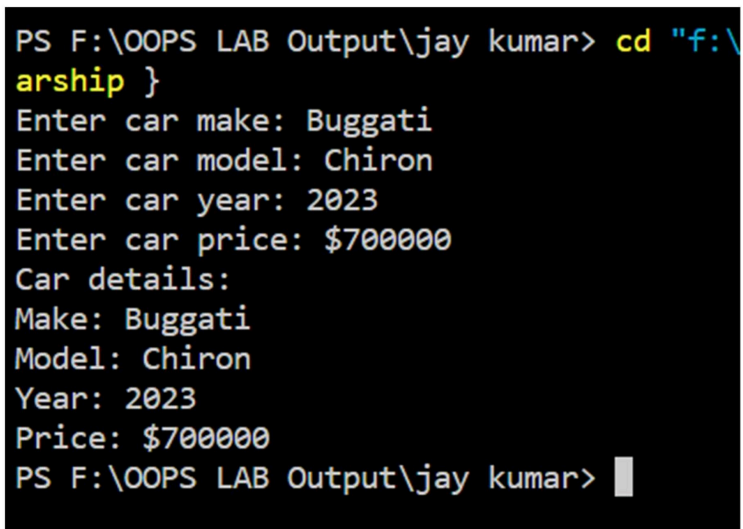
```
double price;

// Input car details from the user
cout << "Enter car make: ";
cin >> make;
cout << "Enter car model: ";
cin >> model;
cout << "Enter car year: ";
cin >> year;
cout << "Enter car price: $";
cin >> price;

// Create a car object using the constructor with user input
Car car(make, model, year, price);

// Display car information
cout << "Car details:" << endl;
car.display();
return 0;
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\arship"
Enter car make: Buggati
Enter car model: Chiron
Enter car year: 2023
Enter car price: $700000
Car details:
Make: Buggati
Model: Chiron
Year: 2023
Price: $700000
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 18

Write a program in c++ using constructor on Order Processing System.

```
#include <iostream>
#include <string>
using namespace std;
class Order {
public:
    int orderNumber;
    string customerName;
    string product;
    int quantity;
    double totalPrice;

    // Constructor to initialize order details
    Order(int number, string name, string prod, int qty, double price) :
    orderNumber(number), customerName(name), product(prod), quantity(qty),
    totalPrice(price) {
    }

    // Function to calculate total price
    void calculateTotalPrice() {
        totalPrice = quantity * totalPrice;
    }

    // Display order information
    void display() {
        cout << "Order Number: " << orderNumber << endl;
        cout << "Customer Name: " << customerName << endl;
    }
};
```



```

        cout << "Product: " << product << endl;
        cout << "Quantity: " << quantity << endl;
        cout << "Total Price: $" << totalPrice << endl;
    }
};

int main() {
    int orderNumber;
    string customerName, product;
    int quantity;
    double price;

    // Input order details from the user
    cout << "Enter Order Number: ";
    cin >> orderNumber;
    cin.ignore(); // Ignore the newline character in the input buffer.
    cout << "Enter Customer Name: ";
    getline(cin, customerName);
    cout << "Enter Product: ";
    getline(cin, product);
    cout << "Enter Quantity: ";
    cin >> quantity;
    cout << "Enter Price per Unit: $";
    cin >> price;

    // Create an order object using the constructor with user input
    Order order(orderNumber, customerName, product, quantity, price);
}

```

```
// Calculate the total price and display order information
order.calculateTotalPrice();

cout << "\nOrder Details:" << endl;
order.display();

return 0;
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "f:
if ($?) { .\OrderProcessingsystem }
Enter Order Number: 4687636
Enter Customer Name: Jay
Enter Product: Moto Edge 40 Neo
Enter Quantity: 1
Enter Price per Unit: $312

Order Details:
Order Number: 4687636
Customer Name: Jay
Product: Moto Edge 40 Neo
Quantity: 1
Total Price: $312
PS F:\OOPS LAB Output\jay kumar> █
```

PRACTICAL NO. 19

Write a program IN C++ to add two number using constructor.

```
#include <iostream>
using namespace std;
class Adder {
public:
    int num1;
    int num2;
    int sum;

    Adder(int n1, int n2) : num1(n1), num2(n2) {
        sum = num1 + num2;
    }
};

int main() {
    int number1, number2;
    cout << "Enter the first number: ";
    cin >> number1;
    cout << "Enter the second number: ";
    cin >> number2;

    Adder adder(number1, number2);
    cout << "Sum: " << adder.sum << endl;
    return 0;
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "f:\OOPS
.\addutionOfTwoNum }
Enter the first number: 87
Enter the second number: 89
Sum: 176
PS F:\OOPS LAB Output\jay kumar> █
```

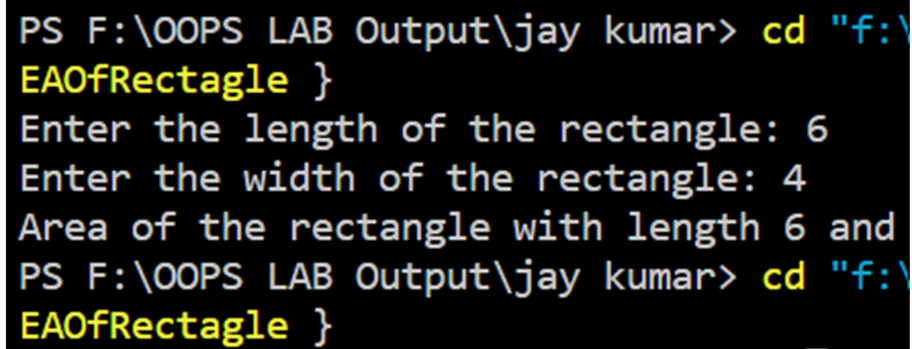
PRACTICAL NO. 20

Write a program in C++ to calculate the area of Rectangle using constructor.

```
#include <iostream>
using namespace std;
class Rectangle {
private:
    double length;
    double width;
public:
    Rectangle(double l, double w) : length(l), width(w) {
    }
    double calculateArea() {
        return length * width;
    }
    double getLength() {
        return length;
    }
    double getWidth() {
        return width;
    }
};
int main() {
    double length, width;
    cout << "Enter the length of the rectangle: ";
    cin >> length;
    cout << "Enter the width of the rectangle: ";
    cin >> width;
```

```
Rectangle rectangle(length, width);  
  
cout << "Area of the rectangle with length " << rectangle.getLength() << " and  
cout<<"and width " << rectangle.getWidth() << " is: " <<  
rectangle.calculateArea() << endl;  
  
return 0;  
}
```

OUTPUT



```
PS F:\OOPS LAB Output\jay kumar> cd "f:\  
EAOfRectagle }  
Enter the length of the rectangle: 6  
Enter the width of the rectangle: 4  
Area of the rectangle with length 6 and  
PS F:\OOPS LAB Output\jay kumar> cd "f:\  
EAOfRectagle }
```

PRACTICAL NO. 21

Write a program in c++ Using constructor to print age and name by user.

```
#include <iostream>
#include <string>
using namespace std;
class Person {
private:
    string name;
    int age;
public:
    Person(string n, int a) {
        name = n;
        age = a;
    }
    void displayInfo() {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << " years" << endl;
    }
};

int main() {
    string name;
    int age;
    cout << "Enter your name: ";
    getline(cin, name); // Allowing spaces in the name
    cout << "Enter your age: ";
    cin >> age;
```

```
Person person(name, age);  
cout << "Person's Information:" << endl;  
person.displayInfo();  
return 0;  
}
```

OUTPUT

```
PS F:\OOPS LAB Output\jay kumar> cd "f  
}  
Enter your name: Jack sparrow  
Enter your age: 43  
Person's Information:  
Name: Jack sparrow  
Age: 43 years  
PS F:\OOPS LAB Output\jay kumar> █
```


PRACTICAL NO. 22

Write a program that demonstrate the use of all types of inheritance.

```
#include <iostream>
using namespace std;
// Base class
class Shape {
public:
    virtual void draw() const {
        cout << "Drawing a shape" << endl;
    }
};
// Single Inheritance
class Circle : public Shape {
public:
    void draw() const override {
        cout << "Drawing a circle" << endl;
    }
};
// Multiple Inheritance
class Rectangle {
public:
    void draw() const {
        cout << "Drawing a rectangle" << endl;
    }
};
class Square : public Shape, public Rectangle {
public:
    void draw() const override {
        Shape::draw(); // Specify the draw() method from the Shape base class
        Rectangle::draw(); // Specify the draw() method from the Rectangle base class
    }
};
// Multilevel Inheritance
```

```

class Triangle : public Shape {
public:
    void draw() const override {
        cout << "Drawing a triangle" << endl;
    }
};

class EquilateralTriangle : public Triangle {
public:
    void draw() const override {
        cout << "Drawing an equilateral triangle" << endl;
    }
};

// Hierarchy Inheritance
class Polygon : public Shape {
public:
    void draw() const override {
        cout << "Drawing a polygon" << endl;
    }
};

class Pentagon : public Polygon {
public:
    void draw() const override {
        cout << "Drawing a pentagon" << endl;
    }
};

class Hexagon : public Polygon {
public:
    void draw() const override {
        cout << "Drawing a hexagon" << endl;
    }
};

// Hybrid Inheritance (combination of multiple, multilevel, and/or hierarchy)
class CustomShape : public Circle, public Square {
public:
    void draw() const override {

```

```

        Circle::draw(); // Specify the draw() method from the Circle base class
        Square::draw(); // Specify the draw() method from the Square base class
    }
};

int main() {
    // Single Inheritance
    Circle circle;
    circle.draw();

    // Multiple Inheritance
    Square square;
    square.draw();

    // Multilevel Inheritance
    EquilateralTriangle triangle;
    triangle.draw();

    // Hierarchy Inheritance
    Pentagon pentagon;
    pentagon.draw();

    Hexagon hexagon;
    hexagon.draw();

    // Hybrid Inheritance
    CustomShape customShape;
    customShape.draw();

    return 0;
}

```

OUTPUT

```

PS D:\My coding\physicswallah C++ with DSA coding files\OOPs\Inheri
++ with DSA coding files\OOPs\Inheritance\" ; if ($?) { g++ all_type
ype }
Drawing a circle
Drawing a shape
Drawing a rectangle
Drawing an equilateral triangle
Drawing a pentagon
Drawing a hexagon
Drawing a circle
Drawing a shape
Drawing a rectangle
PS D:\My coding\physicswallah C++ with DSA coding files\OOPs\Inheri

```

PRACTICAL NO. 23

Write a program of Function Overriding.

```
#include <iostream>
using namespace std;
class Parent {
public:
    void Print()
    {
        cout << "Base Function" << endl;
    }
};
class Child : public Parent {
public:
    void Print()
    {
        cout << "Derived Function" << endl;
    }
};
int main()
{
    Child C;
    C.Print();
    return 0;
}
```

OUTPUT

```
overriding.cpp -o ove
Derived Function
PS D:\My coding\physi
```

PRACTICAL NO. 24

Write a program using abstract class in cpp.

```
#include <iostream>
using namespace std;
class A
{
public:
    virtual void show() = 0;
    void disp()
    {
        cout << "Hi i m Base Class"<<endl;
    }
};
class B : public A
{
public:
    void show()
    {
        cout << "Hi i m Derived Class";
    }
};
int main()
{
    A *P;
    B ob;
    P=&ob;
    P->disp();
    P->show();
}
```

```
++ with DSA coding files\001
act }
Hi i m Base Class
Hi i m Derived Class
PS D:\My coding\physicswalla
```

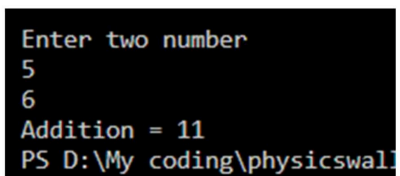
OUTPUT

PRACTICAL NO. 25

Write a program in cpp using Friend function.

```
#include<iostream>
using namespace std;
class demo{
    int a,b;
    public:
    void getdata();
    friend int sum(demo);
};
void demo::getdata()
{
    cout<<"\nEnter two number "<<endl;;
    cin>>a>>b;
}
int sum(demo dd)
{
    return(dd.a+dd.b);
}
int main()
{
    demo dd;
    dd.getdata();
    cout<<"Addition = "<<sum(dd);
    return 0;
}
```

OUTPUT



```
Enter two number
5
6
Addition = 11
PS D:\My coding\physicswall
```