1- Create a program that defines a class Person with private members name, age, and address. Implement public methods setDetails to set person details and displayInfo to display the person's information. and add a public method celebrateBirthday that increments the person's age by 1.

Output

```
Happy Birthday! Age incremented to 26.
Name: John Doe
Age: 26
Address: 123 Main St
```

Solution

```cpp
// www.gammal.tech

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

class Person {
private:
    string name;
    int age;
    string address;

public:
    void setDetails(string n, int a, string addr) {
        name = n;
        age = a;
        address = addr;
    }

    void celebrateBirthday() {
        age++;
        cout << "Happy Birthday! Age incremented to " << age << "." << endl;
    }

    void displayInfo() {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << endl;
        cout << "Address: " << address << endl;
    }
};

int main() {
    Person myPerson;
    myPerson.setDetails("John Doe", 25, "123 Main St");
    myPerson.celebrateBirthday();
    myPerson.displayInfo();
    return 0;
}
```

2- Create a program with a class Bank that has private members balance. Implement public methods setBalance to set the initial balance and withdraw to withdraw an amount. Display the updated balance after each withdrawal.

Output

```
$200 withdrawn successfully. Updated Balance: $800
Invalid withdrawal amount or insufficient balance.
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class Bank {
private:
    double balance;

public:
    void setBalance(double initialBalance) {
        balance = initialBalance;
    }

    void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            cout << "$" << amount << " withdrawn successfully. Updated Balance: $" << balance <<
endl;
        } else {
            cout << "Invalid withdrawal amount or insufficient balance." << endl;
        }
    }
};

int main() {
    Bank myAccount;
    myAccount.setBalance(1000.0);
    myAccount.withdraw(200.0);
    myAccount.withdraw(900.0);
    return 0;
}
```

3- Extend the previous program to add a public method deposit to deposit money into the account and display the updated balance.

توسيع البرنامج السابق لإضافة طريقة إيداع عامة لإيداع الأموال في الحساب وعرض الرصيد المحدث.

Output

```
$200 withdrawn successfully. Updated Balance: $800
$500 deposited successfully. Updated Balance: $1300
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class Bank {
private:
    double balance;

public:
    void setBalance(double initialBalance) {
        balance = initialBalance;
    }

    void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            cout << "$" << amount << " withdrawn successfully. Updated Balance: $" << balance <<
endl;
        } else {
            cout << "Invalid withdrawal amount or insufficient balance." << endl;
        }
    }

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            cout << "$" << amount << " deposited successfully. Updated Balance: $" << balance <<
endl;
        } else {
            cout << "Invalid deposit amount." << endl;
        }
    }
};

int main() {
    Bank myAccount;
    myAccount.setBalance(1000.0);
    myAccount.withdraw(200.0);
    myAccount.deposit(500.0);
    return 0;
}
```

4- Create a program with a class Car that has private members make, model, and year. Implement public methods setDetails to set car details and displayInfo to display the car information.

Output

```
Make: Toyota
Model: Camry
Year: 2022
```

Solution

```cpp
// www.gammal.tech

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

class Car {
private:
    string make;
    string model;
    int year;

public:
    void setDetails(string carMake, string carModel, int carYear) {
        make = carMake;
        model = carModel;
        year = carYear;
    }

    void displayInfo() {
        cout << "Make: " << make << endl;
        cout << "Model: " << model << endl;
        cout << "Year: " << year << endl;
    }
};

int main() {
    Car myCar;
    myCar.setDetails("Toyota", "Camry", 2022);
    myCar.displayInfo();
    return 0;
}
```

5- Extend the previous program to add a public method checkAge that determines if the car is considered old (more than 10 years old) or not.

قم بتوسيع البرنامج السابق لإضافة طريقة checkAge العامة التي تحدد ما إذا كانت السيارة تعتبر قديمة (أكثر من 10 سنوات) أم لا.

Output

```
Make: Toyota
Model: Camry
Year: 2010
This car is considered old.
```

Solution

```cpp
// www.gammal.tech

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

class Car {
private:
    string make;
    string model;
    int year;

public:
    void setDetails(string carMake, string carModel, int carYear) {
        make = carMake;
        model = carModel;
        year = carYear;
    }

    void checkAge() {
        if (2024 - year > 10) {
            cout << "This car is considered old." << endl;
        } else {
            cout << "This car is not considered old." << endl;
        }
    }

    void displayInfo() {
        cout << "Make: " << make << endl;
        cout << "Model: " << model << endl;
        cout << "Year: " << year << endl;
    }
};

int main() {
    Car myCar;
    myCar.setDetails("Toyota", "Camry", 2010);
    myCar.displayInfo();
    myCar.checkAge();
    return 0;
}
```

6- Create a program with a class Book that has private members title, author, and publicationYear. Implement public methods setDetails to set book details and displayInfo to display the book information.

Input

```
Enter book title: The Great Gatsby
Enter author name: F. Scott Fitzgerald
Enter publication year: 1925
```

Output

```
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Publication Year: 1925
```

Solution

```cpp
// www.gammal.tech

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

class Book {
private:
    string title;
    string author;
    int publicationYear;

public:
    void setDetails() {
        cout << "Enter book title: ";
        getline(cin, title);

        cout << "Enter author name: ";
        getline(cin, author);

        cout << "Enter publication year: ";
        cin >> publicationYear;
        cin.ignore();  // Clear the newline character from the input buffer
    }

    void displayInfo() {
        cout << "Title: " << title << endl;
        cout << "Author: " << author << endl;
        cout << "Publication Year: " << publicationYear << endl;
    }
};

int main() {
    Book myBook;
    myBook.setDetails();
    myBook.displayInfo();
    return 0;
}
```

7- Extend the previous program to add a public method isModern that determines if the book is considered modern (published in the last 20 years).

يمتد البرنامج السابق لإضافة الطريقة العامة isModern التي تحدد ما إذا كان الكتاب يعتبر حديثًا (تم نشره في آخر 20 عامًا).

Input

```
Enter book title: The Alchemist
Enter author name: Paulo Coelho
Enter publication year: 1988
```

Output

```
Title: The Alchemist
Author: Paulo Coelho
Publication Year: 1988
This book is not considered modern.
```

Solution

```cpp
// www.gammal.tech

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

class Book {
private:
    string title;
    string author;
    int publicationYear;

public:
    void setDetails() {
        cout << "Enter book title: ";
        getline(cin, title);

        cout << "Enter author name: ";
        getline(cin, author);

        cout << "Enter publication year: ";
        cin >> publicationYear;
        cin.ignore();  // Clear the newline character from the input buffer
    }

    void isModern() {
        int currentYear = 2024;
        if (currentYear - publicationYear <= 20) {
            cout << "This book is considered modern." << endl;
        } else {
            cout << "This book is not considered modern." << endl;
        }
    }

    void displayInfo() {
        cout << "Title: " << title << endl;
        cout << "Author: " << author << endl;
        cout << "Publication Year: " << publicationYear << endl;
    }
};

int main() {
    Book myBook;
    myBook.setDetails();
    myBook.displayInfo();
    myBook.isModern();
    return 0;
}
```

8- Create a program with a class Triangle that has private members base and height. Implement public methods setDimensions to set the base and height, and calculateArea to calculate and display the area of the triangle.

Input

```
Enter base length: 6.0
Enter height length: 8.0
```

Output

```
Area of the triangle: 24
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class Triangle {
private:
    double base;
    double height;

public:
    void setDimensions() {
        cout << "Enter base length: ";
        cin >> base;

        cout << "Enter height length: ";
        cin >> height;
    }

    void calculateArea() {
        double area = 0.5 * base * height;
        cout << "Area of the triangle: " << area << endl;
    }
};

int main() {
    Triangle myTriangle;
    myTriangle.setDimensions();
    myTriangle.calculateArea();
    return 0;
}
```

9- Extend the previous program to add a public method classify that determines if the triangle is equilateral, isosceles, or scalene.

قم بتوسيع البرنامج السابق لإضافة طريقة تصنيف عامة تحدد ما إذا كان المثلث متساوي الأضلاع، أو متساوي الساقين، أو مختلف الأضلاع.

Input

```
Enter base length: 5.0
Enter height length: 5.0
```

Output

```
Area of the triangle: 12.5
This triangle is equilateral.
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class Triangle {
private:
    double base;
    double height;

public:
    void setDimensions() {
        cout << "Enter base length: ";
        cin >> base;

        cout << "Enter height length: ";
        cin >> height;
    }

    void classify() {
        if (base == height) {
            cout << "This triangle is equilateral." << endl;
        } else if (base == height || base != height) {
            cout << "This triangle is isosceles." << endl;
        } else {
            cout << "This triangle is scalene." << endl;
        }
    }

    void calculateArea() {
        double area = 0.5 * base * height;
        cout << "Area of the triangle: " << area << endl;
    }
};

int main() {
    Triangle myTriangle;
    myTriangle.setDimensions();
    myTriangle.calculateArea();
    myTriangle.classify();
    return 0;
}
```

10- Create a program with a class BankAccount that has private members accountNumber, accountHolder, and balance. Implement public methods setDetails to set account details and displayInfo to display the account information

قم بإنشاء برنامج calss BankAccount التي تحتوي على رقم حساب الأعضاء الخاص وحامل الحساب والرصيد. قم بتنفيذ الطرق العامة setDetails لتعيين تفاصيل الحساب وdisplayInfo لعرض معلومات الحساب

Input

```
Enter account number: 12345
Enter account holder name: Jogn Doe
Enter initial balance: $1000
```

Output

```
Account Number: 12345
Account Holder: Jogn Doe
Balance: $1000
```

Solution

```cpp
// www.gammal.tech

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

class BankAccount {
private:
    int accountNumber;
    string accountHolder;
    double balance;

public:
    void setDetails() {
        cout << "Enter account number: ";
        cin >> accountNumber;
        cin.ignore();  // Clear the newline character from the input buffer

        cout << "Enter account holder name: ";
        getline(cin, accountHolder);

        cout << "Enter initial balance: $";
        cin >> balance;
    }

    void displayInfo() {
        cout << "Account Number: " << accountNumber << endl;
        cout << "Account Holder: " << accountHolder << endl;
        cout << "Balance: $" << balance << endl;
    }
};

int main() {
    BankAccount myAccount;
    myAccount.setDetails();
    myAccount.displayInfo();
    return 0;
}
```