**Practical-1**

**Aim: WAP to create a Message class with a constructor that takes a single string with a default value. Create a private member string, and in the constructor simply assign the argument string to your internal string. Create two overloaded member functions called print( ): one that takes no arguments and simply prints the message stored in the object, and one that takes a string argument, which it prints in addition to the internal message.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Message

{

private:

char member[100];

public:

Message(char str[])

{

strcpy(this->member,str);

}

void print()

{

cout << this->member << endl;

}

void print(char ch[])

{

cout << ch << endl;

}

};

int main()

{

Message m("=> Surat is going to become IT hub.");

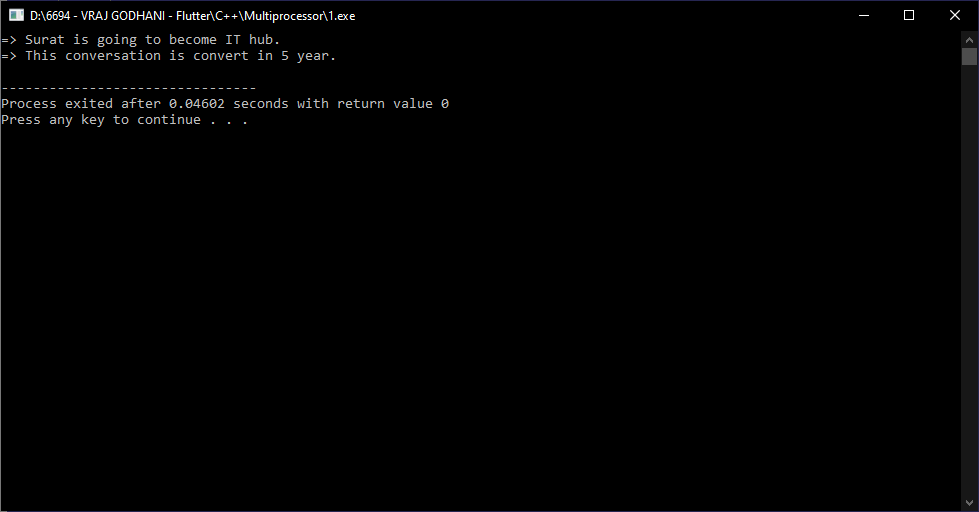
m.print();

m.print("=> This conversation is convert in 5 year.");

return 0;

}

**Output:**

****

**Practical-2**

**Aim: WAP to create a Message class with a constructor that takes a single string with a default value. Create a private member string, and in the constructor simply assign the argument string to your internal string. Create two overloaded member functions called print( ): one that takes no arguments and simply prints the message stored in the object, and one that takes a string argument, which it prints in addition to the internal message.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Vegetable

{

public :

void Eat()

{

cout << "=> I want to eat potato vegetable." << endl;

}

};

class NonVegetable : public Vegetable

{

public :

void Eat()

{

cout << "=> I want to eat Fish fry." << endl;

Vegetable :: Eat();

}

};

int main()

{

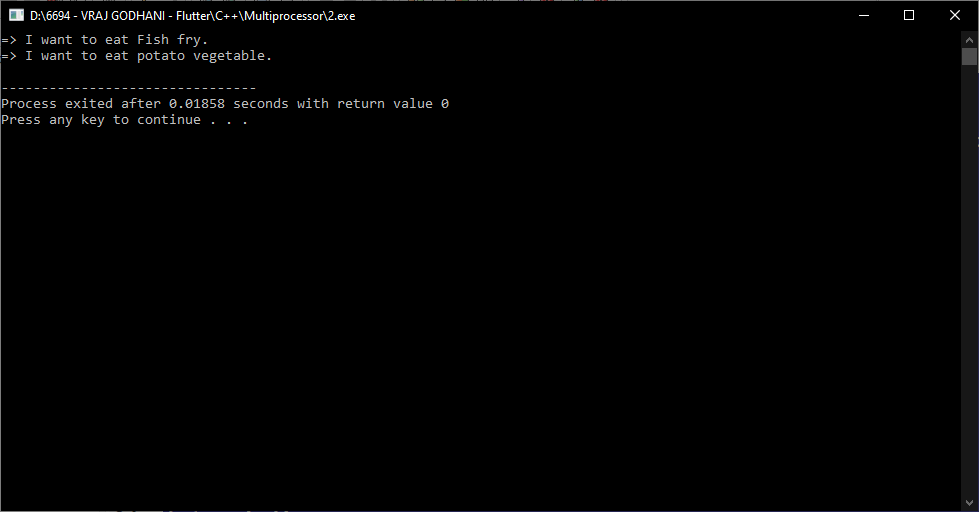
NonVegetable nv;

nv.Eat();

return 0;

}

**Output:**

****