



DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute Affiliated to VTU, Belagavi

Approved by AICTE & ISO 9001 - 2015 Certified)

Accredited by National Assessment and Accreditation Council (NAAC) with 'A' Grade

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru - 560 078.

UG ASSIGNMENT RECORD

ACADEMIC YEAR

Name :

USN :Semester.....

Section :

Branch :

Course Name :

Course code :

Name of the Faculty :

Sl. No.	Date	ASSIGNMENT	
		Max. Marks	Marks Awarded
1		10	

Signature of the Candidate

Signature of the Faculty with Date

Signature of the HOD

DAYANANDA SAGAR COLLEGE OF ENGINEERING

VISION

To impart quality technical education with a focus on Research and Innovation emphasising on Development of Sustainable and Inclusive Technology for the benefit of society

MISSION

- * To provide an environment that enhances creativity and Innovation in pursuit of Excellence.
- * To nurture teamwork in order to transform individuals as responsible leaders and entrepreneurs.
- * To train the students to the changing technical scenario and make them to understand the importance of Sustainable and Inclusive technologies.

QUALITY POLICY

We, at Dayananda Sagar College of Engineering, are committed to continually improving and delivering competitive, quality technical education to the utmost satisfaction of students, their parents and potential employers.



ASSIGNMENT

Assignment Display Date	
Assignment Submission Date	
Return of Assignment Date	

DECLARATION

I bearing the USN / Roll No.....

here by declare that, I have written the assignment on my own referring to the standard text books / reference books & have not copied from any other students.

Signature of the Student

Date

Signature of the Faculty

Marks Awarded (10)

Date

CONTENTS

	Pg.No:
• Overview	4
• Topics covered	5
• Flowchart/Block diagram	6
• C++ Program Code	7
• Output	13
• Conclusion	14
• References	14

IMPLEMENTATION OF VENDING MACHINE USING MEALY FINITE STATE MACHINE IN C++ LANGUAGE

OVERVIEW

A vending machine is a machine that dispenses items such as snacks, beverages, and other goods after a customer inserts currency or a credit card. The vending machine uses a Mealy finite state machine (FSM) to control its operations.

The FSM for a vending machine using Mealy consists of several states including idle, coin accepted, item selected, item dispensed, and out of stock.

The idle state is the initial state of the vending machine, where it is waiting for a customer to insert coins or a credit card. When a coin is inserted, the machine transitions to the coin accepted state and waits for the customer to make a selection.

Once an item is selected, the machine transitions to the item selected state and waits for the customer to insert enough coins to pay for the item. If the customer has inserted enough coins, the machine transitions to the item dispensed state and dispenses the selected item.

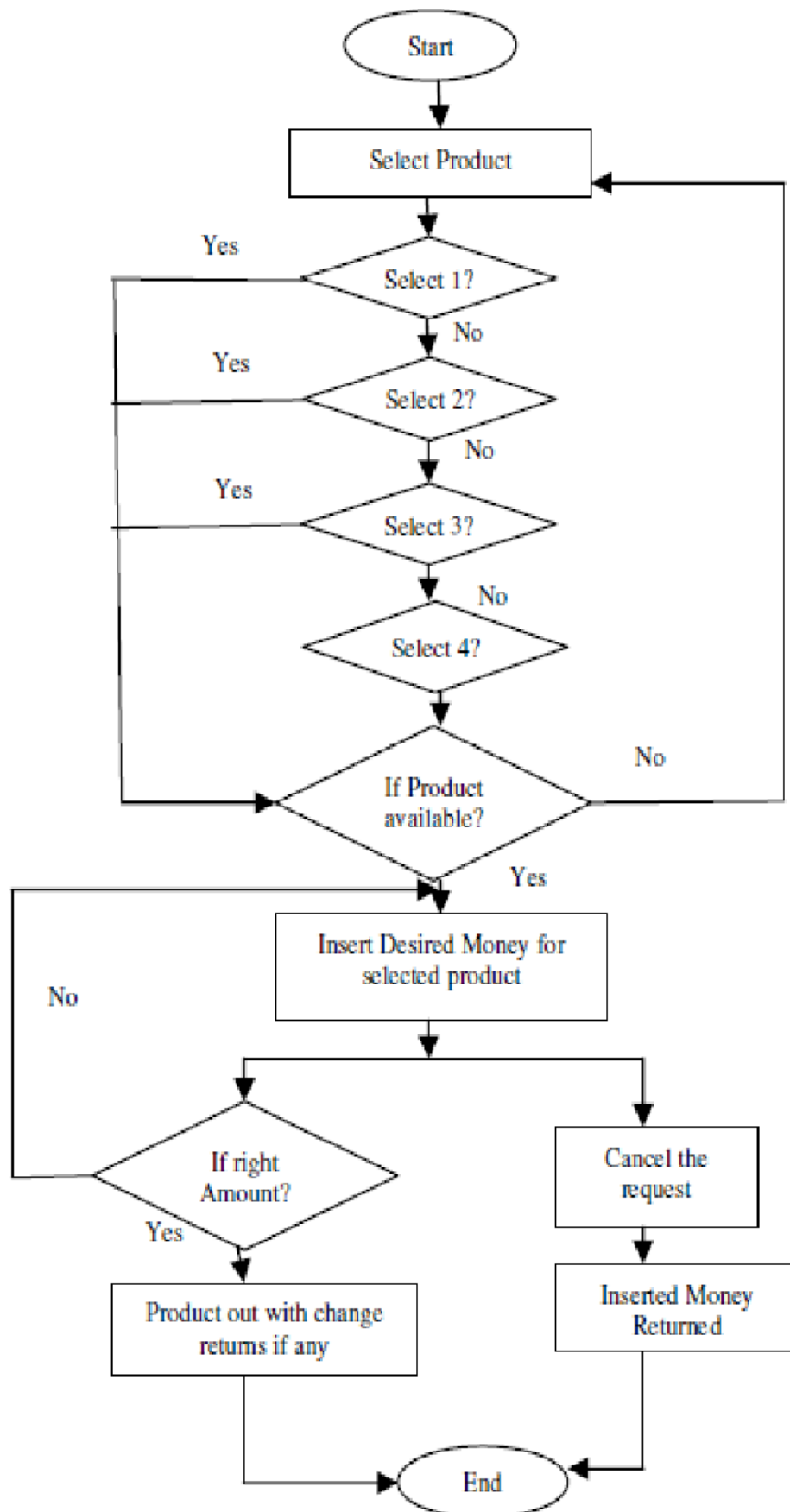
If the selected item is out of stock, the machine transitions to the out-of-stock state and displays a message indicating that the item is not available.

Overall, the FSM for a vending machine using Mealy helps to control the operations of the machine and ensure that it functions correctly and efficiently for customers.

TOPICS COVERED

- Classes
- Structures
- Virtual function
- Constructor
- Pointer
- Function overloading/ function overriding
- Exception handling
- Inheritance

FLOWCHART/BLOCK DIAGRAM



C++ PROGRAM CODE

```
#include<iostream>

#include <string.h>

#include <iomanip>

using namespace std;

struct softdrink
{
    char name[20];
    float price;
    unsigned quantity;
}drink[5];

class vending
{
public:
    virtual void access()
    {
        strcpy(drink[0].name,"Cola"); drink[0].price=0.75; drink[0].quantity=20;
        strcpy(drink[1].name,"Root Beer"); drink[1].price=0.75; drink[1].quantity=20;
        strcpy(drink[2].name,"Lemon Lime"); drink[2].price=0.75; drink[2].quantity=20;
        strcpy(drink[3].name,"Grape Soda"); drink[3].price=0.80; drink[3].quantity=20;
        strcpy(drink[4].name,"Cream Soda"); drink[4].price=0.80; drink[4].quantity=20;
        cout << fixed;
        cout << setprecision(2);
        cout<<" WELCOME TO THE VENDING MACHINE! \n";
        cout<<"~~~~~\n";
```



```

cout<<"Please select your drink: \n";

cout<<" 1) "<<drink[0].name<<"\t\t"<<drink[0].price<<"\t("<<drink[0].quantity<<"
remaining";

cout<<"\n 2) "<<drink[1].name<<"\t\t"<<drink[1].price<<"\t("<<drink[1].quantity<<"
remaining";

cout<<"\n 3) "<<drink[2].name<<"\t\t"<<drink[2].price<<"\t("<<drink[2].quantity<<"
remaining";

cout<<"\n 4) "<<drink[3].name<<"\t\t"<<drink[3].price<<"\t("<<drink[3].quantity<<"
remaining";

cout<<"\n 5) "<<drink[4].name<<"\t\t"<<drink[4].price<<"\t("<<drink[4].quantity<<"
remaining";

cout<<"\n 6) Leave the vending machine \n\n";

}

};

class sell: public vending
{
public:
    sell(): vending() {}

    void run()
    {
        cout<<"\n Choose a drink: ";

        int choice;

        cin>>choice;

        while(choice!=6)
        {
            if(choice >=1 && choice <=5)
            {
                if(drink[choice-1].quantity == 0)

```

```

    {
        cout<<"\n No more " << drink[choice-1].name <<" is available, requires refilling of the
        machine!";

        getchar();

        getchar();

        continue;

    }
}

if(choice == 6)

cout<<"Thank you for using it!";

else if(choice <= 5)

{

float money, price;

cout<<"\n Enter amount inserted: ";

cin>>money;

if(choice>=1 && choice <=3)

{

price = 0.75;

if(money < price)

{

cout<<"\n Enter sufficient amount : ";

getchar();

getchar();

continue;

}

}

else if(choice ==4 || choice ==5)

```

```

{
price = 0.80;
if(money < price)
{
cout<<"\n Enter sufficient amount ";
getchar();
getchar();
continue;
}
}

cout<<"\n Thum, thum, thum, splat..Enjoy your beverage!";
cout<<"\n\nChange calculated: "<< money-price;
drink[choice-1].quantity = drink[choice-1].quantity - 1;
cout<<"\nThere are "<< drink[choice-1].quantity <<" drinks left of that type. \n";
cout<<"Thank you for using it! Have a nice day :D \n";
cout<<"~~~~~\n";
getchar();
getchar();
}
try
{
    if(choice>=6)
        throw 6;
}
catch(int choice)
{
    cout<<"Out of stock!";

```

```
break;
```

```
}
```

```
}
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    vending v;
```

```
    vending *vptr;
```

```
    vptr=&v;
```

```
    vptr->access();
```

```
    sell s;
```

```
    sell *sptr;
```

```
    sptr=&s;
```

```
    sptr->run();
```

```
    return 0;
```

```
}
```

OUTPUT

```
WELCOME TO THE VENDING MACHINE!
-----
Please select your drink:
1) Cola          0.75   (20) remaining
2) Root Beer     0.75   (20) remaining
3) Lemon Lime    0.75   (20) remaining
4) Grape Soda    0.80   (20) remaining
5) Cream Soda    0.80   (20) remaining
6) Leave the vending machine

Choose a drink : 5
Enter amount inserted : 3
Thum, thum, thum, splat..Enjoy your beverage!

Change calculated : 2.20
There are 19 drinks left of that type.
Thank you for using it! Have a nice day :D
-----

Enter amount inserted : 2
Thum, thum, thum, splat..Enjoy your beverage!

Change calculated : 1.20
There are 18 drinks left of that type.
Thank you for using it! Have a nice day :D
-----
```

```
WELCOME TO THE VENDING MACHINE!
~~~
Please select your drink:
1) Cola          0.7500 (20) remaining
2) Root Beer     0.7500 (20) remaining
3) Lemon Lime    0.7500 (20) remaining
4) Grape Soda    0.8000 (20) remaining
5) Cream Soda    0.8000 (20) remaining
6) Leave the vending machine

Choose a drink : 7
out of stock
```

CONCLUSION

Thus, our application of mealy finite state machine using C++, that is the implementation of a vending machine, is successfully automated with the mentioned software requirements. The records that are entered in are automatically updated in the database. The above mentioned testing techniques are also performed and our application is finally ready to be deployed to our customers. This has the advantage of maintaining the records properly without any manual stress. Hence, it becomes easier to view the details later. The maintenance of our system completely depends on the customer's risk.

REFERENCES

- 1) Object Oriented Programming with C++, E. Balaguruswamy, TMH, 6th Edition, 2013.
- 2) Computer Science with C++, Sumita Arora, 2019.
- 3) <https://www.cppbuzz.com/programs/c++/c++-program-of-drinking-machine>
- 4) <https://www.sourcecodesolutions.in/2010/09/petrol-vending-machine-system.html>