

Manual for CS586 14F Project

Feng Huang, A20281629

After executed the program “CS586Project.exe”, it shows:

```
CS586 14F Project
MDA-EFSM-ATM
-----

select ATM-1, ATM-2, or ATM-3
1 = ATM-1
2 = ATM-2
3 = ATM-3
>
```

User can type in a number to select which atm to be created. After type in 1, it shows :

```
MDA-EFSM-ATM
-----

select ATM-1, ATM-2, or ATM-3
1 = ATM-1
2 = ATM-2
3 = ATM-3
>1
you selected ATM-1
ATM-1 created
    Menu of Operations
    1.card(int,string)
    2.pin(string)
    3.deposit(int)
    4.withdraw(int)
    5.balance()
    6.lock(string)
    7.unlock(string)
    8.exit()
    9.quit the demo
Select Operation :
>
```

Now user can select an operation to perform by choosing the number and fill required arguments.
As an example, now perform 1.card(int,string), it shows:

```

5.balance()
6.lock(string)
7.unlock(string)
8.exit()
9.quit the demo
Select Operation :
>1
selected : card(int x,string y)
define balance x :
>100
define pin y:
>abc
ATM-1: please input pin ...
Menu of Operations
1.card(int,string)
2.pin(string)
3.deposit(int)
4.withdraw(int)
5.balance()
6.lock(string)
7.unlock(string)
8.exit()
9.quit the demo
Select Operation :
>

```

In case of mal-input, this parser program will filter invalid input and ask user to re-enter value. As an example now give deposit argument as 'abc', it detects the invalid argument and ask for re-type:

```

Select Operation :
>3
selected : deposit(int d)
define amount d :
>abc
input int wrong
define amount d :
>aaa
input int wrong
define amount d :
>20
ATM-1: deposit 20 to balance 100 , penalty=10
Menu of Operations
1.card(int,string)
2.pin(string)
3.deposit(int)
4.withdraw(int)
5.balance()
6.lock(string)
7.unlock(string)
8.exit()
9.quit the demo
Select Operation :
>

```

Besides, the parser also truncate for int number if user input is float, for instance user input 40.5 for int argument, only 40 will be passed in.

For better observing output from actions (output processor), user can just focus on lines starting with : ATM-n : , because only output in OP starts with that. Other than that are printed by parser (driver).

How to compile the code

This project was coded in Dev-C++, besides the source code in src, a bundle of Dev-C++ project is also provided in folder “src_DevC++”, one can use corresponding IDE to run it.

Also, if you use Linux, for just compile and run it (no makefile), you can use the following command :

```
fenghuang@ubuntu:~/cs586$ ls
ATM.cpp  Data_Store.h  MDA_EFSM.cpp  OP.cpp  Parser.cpp  State.cpp
ATM.h    main.cpp      MDA_EFSM.h   OP.h    Parser.h    State.h
fenghuang@ubuntu:~/cs586$ g++ *.cpp -o demo
fenghuang@ubuntu:~/cs586$ ./demo
      CS586 14F Project
      MDA-EFSM-ATM
-----

select ATM-1, ATM-2, or ATM-3
1 = ATM-1
2 = ATM-2
3 = ATM-3
> |
```

Finally, if you have VS IDE, as VS2010 bundle is also provided, in folder “src_vs”.