

Week 3 Assessment

Generated by Doxygen 1.9.8

1 Write a program for ATM withdrawal	1
1.1 Compile	1
1.2 Run	1
1.3 Generate documentation	1
1.4 Generate pdf documentation	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 customer Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Data Documentation	7
4.1.2.1 balance	7
4.2 denominations Struct Reference	7
4.2.1 Detailed Description	8
4.2.2 Member Data Documentation	8
4.2.2.1 _100	8
4.2.2.2 _200	8
4.2.2.3 _50	8
4.2.2.4 _500	8
4.2.2.5 total_money	8
5 File Documentation	9
5.1 inc/main.h File Reference	9
5.1.1 Typedef Documentation	9
5.1.1.1 customer_t	9
5.1.1.2 denominations_t	10
5.1.2 Function Documentation	10
5.1.2.1 balance_enquiry()	10
5.1.2.2 deposit()	10
5.1.2.3 withdraw()	10
5.2 main.h	11
5.3 README.md File Reference	11
5.4 src/balance.c File Reference	11
5.4.1 Function Documentation	11
5.4.1.1 balance_enquiry()	11
5.5 src/deposit.c File Reference	12
5.5.1 Function Documentation	12
5.5.1.1 deposit()	12
5.6 src/main.c File Reference	12

5.7 src/withdraw.c File Reference	12
5.7.1 Function Documentation	13
5.7.1.1 withdraw()	13
Index	15

Chapter 1

Write a program for ATM withdrawal

- At start it enable deposit where you have to deposit money with the denomination of 500, 200, 100, 50. Once deposited it should display the available total balance and the denomination details. (This should happen only one time)
- Then a menu should show to withdraw, Balance enquiry, Exit
- when the withdrawal amount was entered it should check whether it is a valid input? , Balance is available in account? Amount is within the available denominations
- After withdrawal it should show the available total balance alone.
- Proper denomination is calculated based on the availability and decremented from the total denominations
- when balance enquiry menu is selected display the balance total amount and the available denomination
- when exit is selected display "Thanks for visiting " and terminate the program

1.1 Compile

```
make
```

1.2 Run

```
make run
```

1.3 Generate documentation

```
doxygen Doxyfile
```

1.4 Generate pdf documentation

```
doxygen Doxyfile
cd docs/latex
make
mv refman.pdf ../../
make clean
cd ../../
```


Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

customer	Structure representing a customer with their balance	7
denominations	Structure representing available denominations in the ATM	7

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

inc/ main.h	9
src/ balance.c	11
src/ deposit.c	12
src/ main.c	12
src/ withdraw.c	12

Chapter 4

Class Documentation

4.1 customer Struct Reference

Structure representing a customer with their balance.

```
#include <main.h>
```

Public Attributes

- int [balance](#)

4.1.1 Detailed Description

Structure representing a customer with their balance.

4.1.2 Member Data Documentation

4.1.2.1 balance

```
int customer::balance
```

Balance in the customer's account

The documentation for this struct was generated from the following file:

- inc/[main.h](#)

4.2 denominations Struct Reference

Structure representing available denominations in the ATM.

```
#include <main.h>
```

Public Attributes

- [int _100](#)
- [int _200](#)
- [int _500](#)
- [int _50](#)
- [int total_money](#)

4.2.1 Detailed Description

Structure representing available denominations in the ATM.

4.2.2 Member Data Documentation

4.2.2.1 _100

```
int denominations::_100
```

Number of 100 denomination notes

4.2.2.2 _200

```
int denominations::_200
```

Number of 200 denomination notes

4.2.2.3 _50

```
int denominations::_50
```

Number of 50 denomination notes

4.2.2.4 _500

```
int denominations::_500
```

Number of 500 denomination notes

4.2.2.5 total_money

```
int denominations::total_money
```

Total money in the ATM

The documentation for this struct was generated from the following file:

- [inc/main.h](#)

Chapter 5

File Documentation

5.1 inc/main.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct `denominations`
Structure representing available denominations in the ATM.
- struct `customer`
Structure representing a customer with their balance.

Typedefs

- typedef struct `denominations` `denominations_t`
Structure representing available denominations in the ATM.
- typedef struct `customer` `customer_t`
Structure representing a customer with their balance.

Functions

- void `deposit` (`denominations_t` *denom, `customer_t` *cust)
Function to deposit money into the ATM.
- void `withdraw` (`denominations_t` *denom, `customer_t` *cust)
Function to withdraw money from the ATM.
- void `balance_enquiry` (`denominations_t` *denom, `customer_t` *cust)
Function to check balance and display available denominations.

5.1.1 Typedef Documentation

5.1.1.1 `customer_t`

```
typedef struct customer customer_t
```

Structure representing a customer with their balance.

5.1.1.2 denominations_t

```
typedef struct denominations denominations_t
```

Structure representing available denominations in the ATM.

5.1.2 Function Documentation

5.1.2.1 balance_enquiry()

```
void balance_enquiry (
    denominations_t * denom,
    customer_t * cust )
```

Function to check balance and display available denominations.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

5.1.2.2 deposit()

```
void deposit (
    denominations_t * denom,
    customer_t * cust )
```

Function to deposit money into the ATM.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

5.1.2.3 withdraw()

```
void withdraw (
    denominations_t * denom,
    customer_t * cust )
```

Function to withdraw money from the ATM.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

5.2 main.h

[Go to the documentation of this file.](#)

```
00001 #ifndef MAIN_H
00002 #define MAIN_H
00003
00007 typedef struct denominations {
00008     int _100;
00009     int _200;
00010     int _500;
00011     int _50;
00012     int total_money;
00013 } denominations_t;
00014
00018 typedef struct customer {
00019     int balance;
00020 } customer_t;
00021
00022 // Function declarations
00023 void deposit(denominations_t *denom, customer_t *cust);
00024 void withdraw(denominations_t *denom, customer_t *cust);
00025 void balance_enquiry(denominations_t *denom, customer_t *cust);
00026
00027 #endif // MAIN_H
```

5.3 README.md File Reference

5.4 src/balance.c File Reference

```
#include <stdio.h>
#include "main.h"
Include dependency graph for balance.c:
```

Functions

- void [balance_enquiry](#) ([denominations_t](#) *denom, [customer_t](#) *cust)
Function to check balance and display available denominations.

5.4.1 Function Documentation

5.4.1.1 balance_enquiry()

```
void balance_enquiry (
    denominations_t * denom,
    customer_t * cust )
```

Function to check balance and display available denominations.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

5.5 src/deposit.c File Reference

```
#include <stdio.h>
#include "main.h"
Include dependency graph for deposit.c:
```

Functions

- void [deposit](#) ([denominations_t](#) *denom, [customer_t](#) *cust)
Function to deposit money into the ATM.

5.5.1 Function Documentation

5.5.1.1 deposit()

```
void deposit (
    denominations\_t * denom,
    customer\_t * cust )
```

Function to deposit money into the ATM.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

5.6 src/main.c File Reference

```
#include <stdio.h>
#include "main.h"
Include dependency graph for main.c:
```

5.7 src/withdraw.c File Reference

```
#include <stdio.h>
#include "main.h"
Include dependency graph for withdraw.c:
```

Functions

- void [withdraw](#) ([denominations_t](#) *denom, [customer_t](#) *cust)
Function to withdraw money from the ATM.

5.7.1 Function Documentation

5.7.1.1 withdraw()

```
void withdraw (  
    denominations_t * denom,  
    customer_t * cust )
```

Function to withdraw money from the ATM.

Parameters

<i>denom</i>	Pointer to denominations structure
<i>cust</i>	Pointer to customer structure

Index

- [_100](#)
 - [denominations](#), [8](#)
 - [_200](#)
 - [denominations](#), [8](#)
 - [_50](#)
 - [denominations](#), [8](#)
 - [_500](#)
 - [denominations](#), [8](#)
- [balance](#)
 - [customer](#), [7](#)
- [balance.c](#)
 - [balance_enquiry](#), [11](#)
- [balance_enquiry](#)
 - [balance.c](#), [11](#)
 - [main.h](#), [10](#)
- [customer](#), [7](#)
 - [balance](#), [7](#)
- [customer_t](#)
 - [main.h](#), [9](#)
- [denominations](#), [7](#)
 - [_100](#), [8](#)
 - [_200](#), [8](#)
 - [_50](#), [8](#)
 - [_500](#), [8](#)
 - [total_money](#), [8](#)
- [denominations_t](#)
 - [main.h](#), [9](#)
- [deposit](#)
 - [deposit.c](#), [12](#)
 - [main.h](#), [10](#)
- [deposit.c](#)
 - [deposit](#), [12](#)
- [inc/main.h](#), [9](#), [11](#)
- [main.h](#)
 - [balance_enquiry](#), [10](#)
 - [customer_t](#), [9](#)
 - [denominations_t](#), [9](#)
 - [deposit](#), [10](#)
 - [withdraw](#), [10](#)
- [README.md](#), [11](#)
- [src/balance.c](#), [11](#)
- [src/deposit.c](#), [12](#)
- [src/main.c](#), [12](#)
- [src/withdraw.c](#), [12](#)
- [total_money](#)
 - [denominations](#), [8](#)
- [withdraw](#)
 - [main.h](#), [10](#)
 - [withdraw.c](#), [13](#)
- [withdraw.c](#)
 - [withdraw](#), [13](#)
- [Write a program for ATM withdrawal](#), [1](#)