Account Header:

```
#ifndef ACCOUNT_H
#define ACCOUNT_H
#include <string>

using namespace std;

class Account {

public:
  // Default constructor
  Account();
  // Primary constructor to set elements
  // @params take in first name, last name, and an initial balance
  Account(string first, string last, double balance);
  // Deconstructor to delete the object
  ~Account();
  // Getter functions to access the private member that do not override the private members
  const string getFirstName();
  const string getLastName();
  const double getCurrentBalance();
  // Setter functions to allow new value to be passed to the private members
  void setFirstName(string first);
  void setLastName(string last);
  void setCurrentBalance(double amount);
  // Allows for adding to the current balance
  void debitFrom(double add);
  // Allows for subtracting from the current balance
  void creditTo(double minus);
  // Prints out Account information: first name, last name, current balance
  void print();

private:
  string firstName;
  string lastName;
  double currentBalance;
};

#endif
```

Account Class:

```
#include "account.h"
#include <iostream>
#include <string>

using namespace std;

Account::Account(){
  firstName = "";
  lastName = "";
  currentBalance = 0.00;
};

Account::Account(string first, string last, double balance){
   firstName = first;
   lastName = last;
   currentBalance = balance;
};
Account::~Account(){
  cout << "Deleting Account Object" << endl;
};
const string Account::getFirstName(){
  return firstName;
};
const string Account::getLastName(){
  return lastName;
};
const double Account::getCurrentBalance(){
  return currentBalance;
};
void Account::setFirstName(string first){
  firstName = first;
};
void Account::setLastName(string last){
  lastName = last;
};
void Account::setCurrentBalance(double amount){
  currentBalance = amount;
};
void Account::debitFrom(double add){
  setCurrentBalance(getCurrentBalance() + add);
};
void Account::creditTo(double minus){
  setCurrentBalance(getCurrentBalance() - minus);
};
void Account::print(){
  cout << "First Name: " << getFirstName() << endl;
  cout << "Last Name: " << getLastName() << endl;
  cout << "Current Balance: " << getCurrentBalance() << endl;
};
```

Account Test:

```
#include <iostream>
#include <string>
#include <new>
#include "account.h"

using namespace std;

template<class T>
void objSwap(T& one, T& two){
  T tmp(one);
  one = two;
  two = tmp;
}

int main(){

  // Part A
  cout << "===== PART A =====" << endl;
  Account accountOne("John", "Smith", 100.00);
  accountOne.print();
  cout << "===== END =====" << endl;

  // Part C

  cout << "===== Part C =====" << endl;
  const int SIZE = 10;
  int user = 0;
  int option = 0;
  double amount = 0.00;
  Account * arr[SIZE];
  Account * accountTwo;
  accountTwo = new Account("John", "Smith", 100.00);
  accountTwo -> debitFrom(100.00);
  cout << "===== DEMO debit =====" << endl;
  accountTwo -> print();
  accountTwo -> creditTo(25.00);
  cout << "===== DEMO credit =====" << endl;
  accountTwo -> print();


  // example of how to delete a pointer
  delete accountTwo;

  // Initialize array of pointers
```

```
cout << "===== Initialize Array of pointers =====" << endl;

for(int i = 0; i < SIZE; i++){
  arr[i] = new Account("John", "Smith", 100.00);
  arr[i] -> print();
}

cout << "===== END Initializing Array of pointers" << endl;

// ATM Logic
// User names are based off of the the array generated above
cout << "Please enter your username: " << endl;
cin >> user;


// Allow for multiple actions to occur
while (option != 4) {
  cout << "Please select and option below: " << endl;
  cout << "1. View Balance" << endl;
  cout << "2. Withdraw" << endl;
  cout << "3. Deposit" << endl;
  cout << "4. Exit" << endl;
  cin >> option;
  if(option == 1){
    arr[user] -> print();
  } else if (option == 2){
    cout << "How much would you like to withdraw?" << endl;
    cin >> amount;
    arr[user] -> creditTo(amount);
  } else if (option == 3){
    cout << "How much would you like to deposit?" << endl;
    cin >> amount;
    arr[user] -> debitFrom(amount);
  } else {
    if(option == 4){
      cout << "Thanks for using NU ATM, have a great day!!!" << endl;
    } else {
      cout << "Invalid Option, Try Again..." << endl;
    }
  }
}

cout << "===== END =====" << endl;
cout << "===== Part C =====" << endl;
// Swap Accounts

Account swapOne("ONE", "ONE", 100.00);
Account swapTwo("TWO", "TWO", 200.00);
```

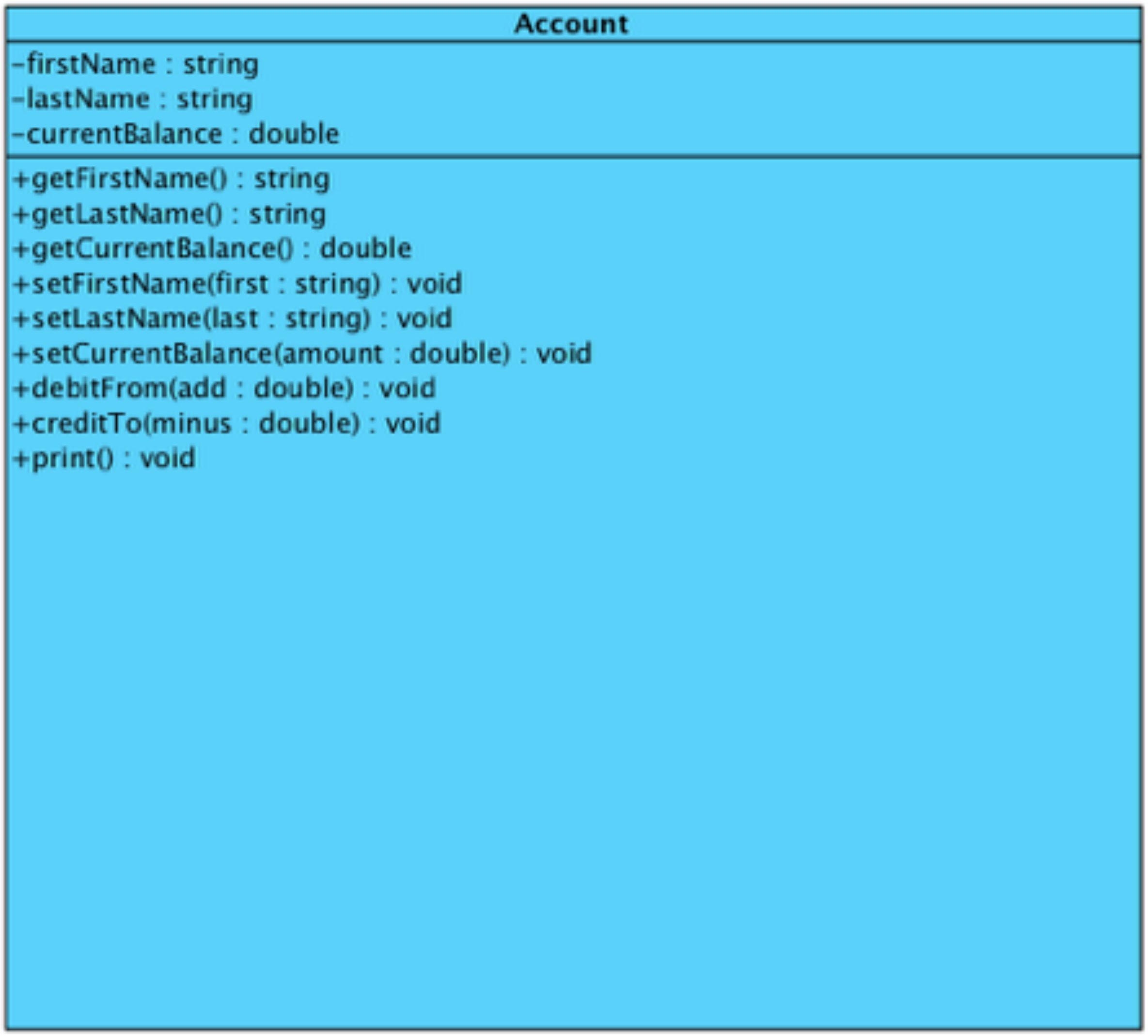
```
  cout << "===== Show initial values before swapping =====" << endl;
  swapOne.print();
  swapTwo.print();

  objSwap(swapOne, swapTwo);

  cout << "===== Show new values after swapping =====" << endl;
  swapOne.print();
  swapTwo.print();

  cout << "===== END =====" << endl;

  // Delete array pointers
  for(int n = 0; n < SIZE; n++){
    delete arr[n];
  }

  // all other objects remaining are deleted by the deconstructor

  return 0;

};
```

Part A:

Q: Show how you would create constant members functions in this class – what is the benefit of such member functions.

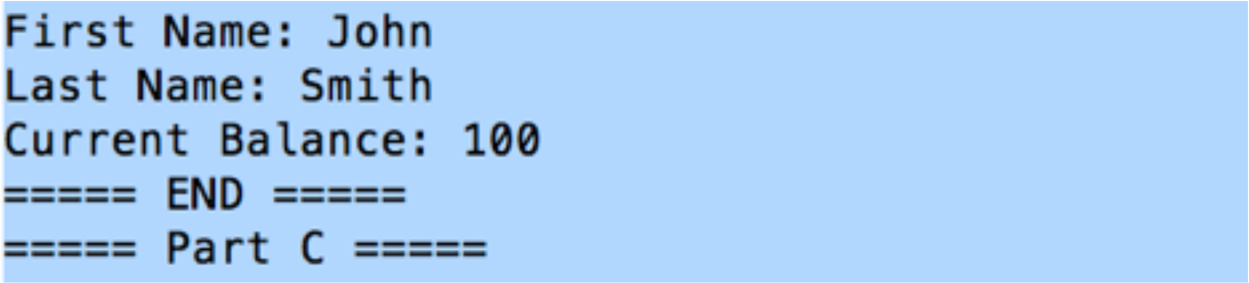
A: See code for how it is implemented. The benefit of using member functions that are constants is that it ensures the value of what is being returned can not be changed in any way.

Part B:
Screen Shots of console output:



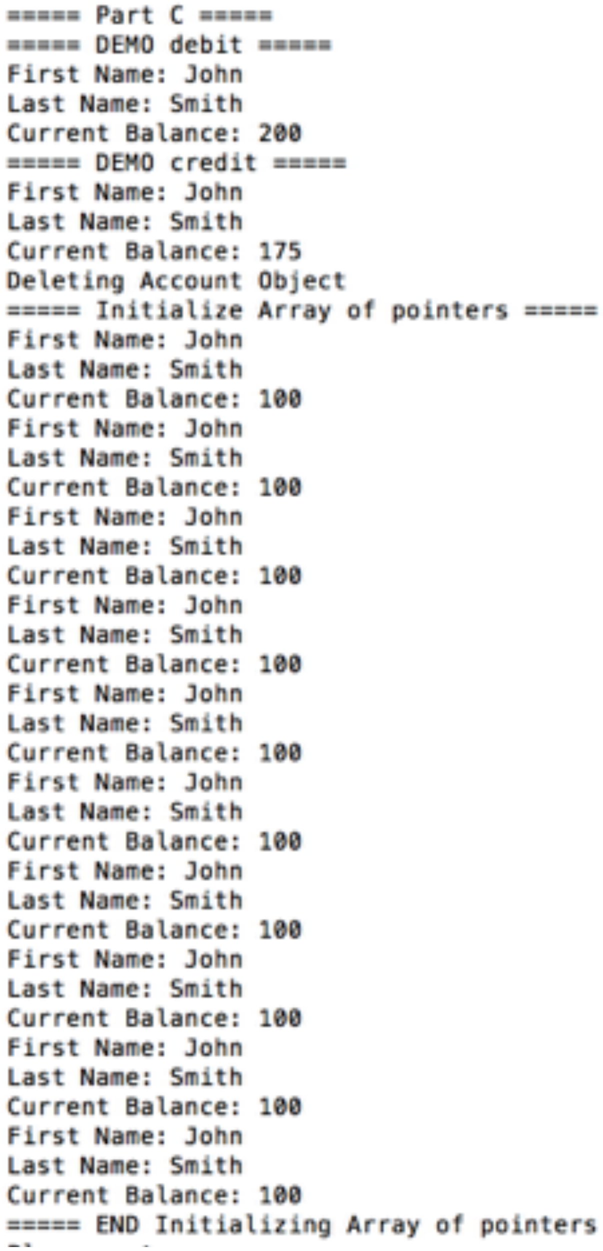
```
Account
-firstName : string
-lastName : string
-currentBalance : double
+getFirstName() : string
+getLastName() : string
+getCurrentBalance() : double
+setFirstName(first : string) : void
+setLastName(last : string) : void
+setCurrentBalance(amount : double) : void
+debitFrom(add : double) : void
+creditTo(minus : double) : void
+print() : void
```

Part A

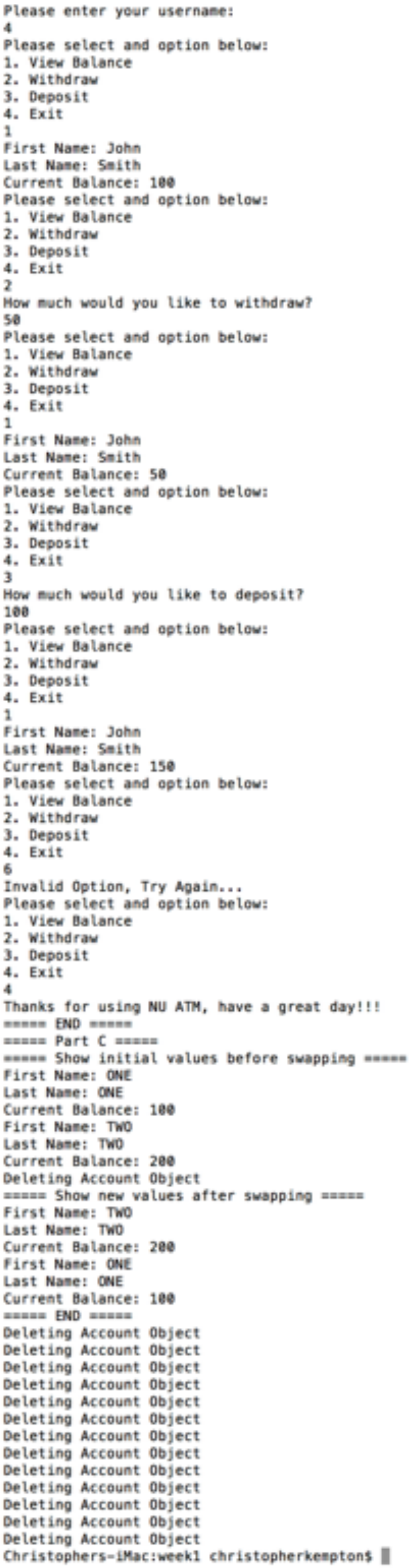


```
First Name: John
Last Name: Smith
Current Balance: 100
===== END =====
===== Part C =====
```

Part C:



```
===== Part C =====
===== DEMO debit =====
First Name: John
Last Name: Smith
Current Balance: 200
===== DEMO credit =====
First Name: John
Last Name: Smith
Current Balance: 175
Deleting Account Object
===== Initialize Array of pointers =====
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
First Name: John
Last Name: Smith
Current Balance: 100
===== END Initializing Array of pointers
```



```
Please enter your username:
4
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
1
First Name: John
Last Name: Smith
Current Balance: 100
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
2
How much would you like to withdraw?
50
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
1
First Name: John
Last Name: Smith
Current Balance: 50
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
3
How much would you like to deposit?
100
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
1
First Name: John
Last Name: Smith
Current Balance: 150
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
6
Invalid Option, Try Again...
Please select and option below:
1. View Balance
2. Withdraw
3. Deposit
4. Exit
4
Thanks for using NU ATM, have a great day!!!
===== END =====
===== Part C =====
===== Show initial values before swapping =====
First Name: ONE
Last Name: ONE
Current Balance: 100
First Name: TWO
Last Name: TWO
Current Balance: 200
Deleting Account Object
===== Show new values after swapping =====
First Name: TWO
Last Name: TWO
Current Balance: 200
First Name: ONE
Last Name: ONE
Current Balance: 100
===== END =====
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Deleting Account Object
Christophers-iMac:week1 christopherkempton$ 
```