

Name _____

1. Write a program that defines an *CheckingAccount* structure containing the member variables

```
char* name
float balance
```

and the member functions

```
bool open(char* aName)
bool setStartBalance(float anAmount)
bool deposit(float anAmount)
bool withdraw(float anAmount)
```

Declare an instance of the *CheckingAccount* structure.

Declare a pointer, *acctPtr*, to such a structure.

Allocate a *CheckingAccount* structure dynamically and assign the address to *acctPtr*.

Demonstrate your account structure design by executing the member functions of the two instances. Interact with the first instance directly and the second instance via the pointer.

Be certain to delete the dynamic instance before exiting your program.

2. Write a simple program to aid in balancing a checkbook. The program's input is single-letter commands followed by an amount. The legal commands are d (deposit), w (withdrawal), and b (set starting balance). The program should print the balance after each transaction. Make sure your program is well behaved even when the input is in error.
3. Write a simple program to model a bank. The bank is to be implemented as a dynamic array of pointers to *CheckingAccount* structures. The bank must implement functions to open an account or close an account. As an account is opened or closed, the size of the array should expand or shrink. The bank should also support a function that will display the name and balance for each customer.