**Pseudocode of method implementation**

**Add\_Account Pseudocode: (Banking**\_**Application.cpp)**

1. Ask for user’s name
2. Provide type of account menu to user
3. **if** acct\_type is savings
4. Set acct\_tye\_str to savings
5. **else** set acct\_tye\_str to checking
6. Attempt to add the account
7. **if** user is new
8. State user is new
9. Get users address
10. Get users telephone number
11. Get users age
12. Provide type of customer menu to user
13. **if** user is an adult
14. Set cust\_type\_str to adult
15. **else if** user is a senior
16. Set cust\_type\_str to senior
17. **else** Set cust\_type\_str to student
18. Add new user’s account
19. **if** new users account was added
20. Display the users new account ID
21. **else** Display that an account was unable to be created

**make\_deposit() Pseudocode: (Bank.h)**

1. Find account
2. **if** account was found
3. deposit the amount specified
4. **else** state the account was unable to be found

**make\_deposit() Pseudocode: (Banking**\_**Application.cpp)**

1. Ask user for account ID
2. Ask user for amount that is being deposited
3. Make deposit

**make\_withdrawal() Pseudocode: (Banking**\_**Application.cpp)**

1. Ask user for account ID
2. Ask user for amount that is being withdrawn
3. Make withdrawal

**add\_account(Customer \*cust, string account\_type) Pseudocode: (Bank.h)**

1. Find the customer
2. **if** the customer is a senior and Account is checking
3. Set acct equal to a new senior checking account
4. **else if** the customer is a senior and Account is savings
5. Setacct equal to a new senior savings account
6. **else if** the customer is an adult and Account is checking
7. Set acct equal to a new adult checking account
8. **else if** the customer is an adult and Account is savings
9. Set acct equal to a new adult savings account
10. **else if** the customer is a student and Account is checking
11. Set acct equal to a new student checking account
12. **else if** the customer is a student and Account is savings
13. Set acct equal to a new student savings account
14. **else** state an error occurred and return that the task was unable to be preformed
15. Return the new account to the user

**add\_account(string name, string account\_type) Pseudocode: (Bank.h)**

1. Find the customer
2. **if** the customer was not found
3. return task was unable to be preformed
4. Return the customer and account\_type as a new account via add\_account

**add\_account(string name, string address, string telephone, int age, string cust\_type, string account\_type) Pseudocode: (Bank.h)**

1. Find customer
2. **if** the customer is a senior
3. Create a senior customer object
4. **else if** the customer is an adult
5. Create an adult customer object
6. **else if** the customer is a student
7. Create a student customer object
8. **else** state an error occurred and return that the task was unable to be preformed
9. Add the customer object to vector of customers
10. Return a new account via add\_account with the arguments cust and account\_type

**Get\_account() Pseudocode: (Bank.h)**

1. **for** each account
2. **if** account has same account number as the one given
3. return the account info to the user
4. If the account was not found, then return that the task was unable to be preformed