**[Banking Part1 Cevans]**

Table of Contents

Cover Page 1

Table of Contents 2

Statement of Independent effort 3

Analysis of Specifications 4

Pseudocode: 4

Flowchart: 4

Test Cases 6

Code 9

Grade sheet 14

# Statement of Independent Effort

I, [Charles, Evans], hereby certify that is my original work completed without the assistance of anyone or any outside resources.

OR

I, [Charles, Evans], hereby certify that this is my original work completed with the assistance of [NAME]/the resources listed in the reference. I used these resources in the following areas: […].

Charles Evans

[YOUR NAME]

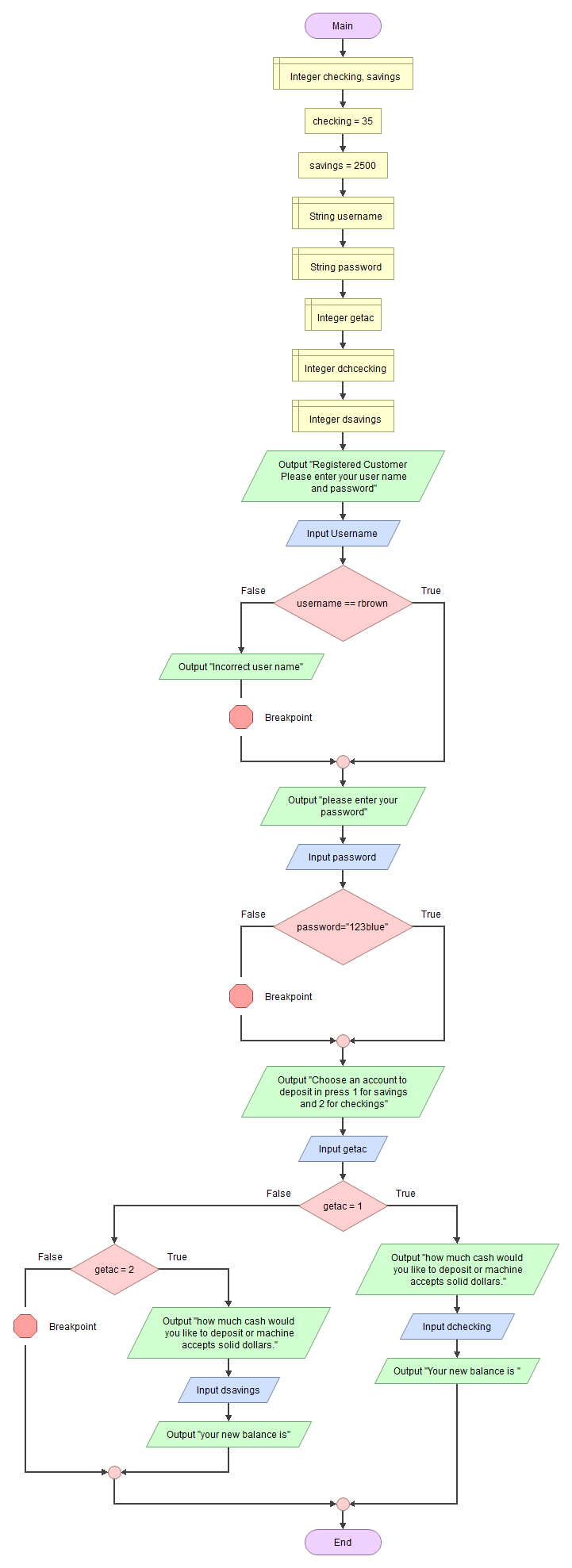
# Analysis of Specifications:

For this program I will need several inputs both to read in information to user and to walk them through the process of using the ATM. I also want to remember as per the guidelines to effectively document my work so other coders can easily understand it. My basic goal is to create a program that first ask for username and password then uses integer variable to increase eithers savings variable or checking variable.

Pseudocode :

1. Declare variable for checkings and savings as well as passwords.
2. Get password and username from user.
3. Verify username and password.
4. Give the user an option to choose a number for checking or savings.
5. Verify checkings or savings.
6. Add user input to amount in checking or savings.
7. 7. Print out new balance to user.

Flowchart:



Test Cases:

Test Case1. //This is an example of user entering wrong User name//

Output: registered customer please enter your username and password”

User Input: “rblue” Output: “Wrong username Buddy”. Terminate.

Test Case 2. This is another example of user entering wrong Username.

User input : “rblack” Output: “Wrong username Buddy”. Terminate.

Test Case 3. This is also an example of user entering wrong password

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue4” Output: “Incorrect password”

Test Case 4. This is another example of user entering wrong password

User input: “rblue”. Coutput: “Now can you please enter your password for me” User input: “brownjohn” Output: “Incorrect password”. Terminate.

Test Case 5. This is an example of user successfully depositing in savings account.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “1” Coutput: "How much cash would you like to deposit dear customer our machine accepts only solid dollar integers" Input: “100” Output: “Your new balance is Savings:2600 Checking 35”

Test Case 6. This is an example of user successfully depositing in savings account.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “1” Coutput: "How much cash would you like to deposit dear customer our machine accepts only solid dollar integers" Input: “99998888” Output: “Your new balance is Savings:100001388 Checking 35”

Test Case 7: This is an example of user successfully depositing into checking account.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “2” Coutput: "how much cash mr brown" Input: “145” Output: “Your new balance is Savings: Checking 180 Savings: 2500”

Test Case 8. This is an example of user successfully depositing into checking account.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “2” Coutput: "how much cash mr brown" Input: “145” Output: “Your new balance is Savings: Checking : 55 Savings: 2500”.

Test Case 9. This is an test case to see what happens when the user enters a decimal representing change.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “2” Coutput: "how much cash mr brown" Input: “.9” Output: “no pennies sir”. Terminate program.

Test Case 10. This is a test case to see what happens when the user enters pennies or non integer values.

User input : “rblue”. Coutput: “Now can you please enter your password for me” User input: “blue123” Output: "Now in order to make your deposit will you please press 1 for savings and 2 for checking " User input: “1” Coutput: ""How much cash would you like to deposit dear customer our machine accepts only solid dollar integers"" Input: “30.3” Output: “Once again No pennies”.

1. User input:
2. User input:
3. User input:
4. User input:
5. User input:
6. User input:
7. User input:
8. User input

Code:

#include<iostream>

#include<string>

#include<fstream>

#include<iomanip>

using namespace std;

int main()

{

int checking;

int savings;

string username;

string password;

int getac;

int dchecking;

int dsavings;

checking = 35;

savings = 2500;

cout <<"registered Customer please enter your username and password " <<endl;

cin >>username;

if (username=="rbrown") {

cout <<"good job"<<endl;

cout <<"Now, Can you please enter your password for me " << endl;

cin >>password;

if (password=="blue123") {

cout <<"nice work" <<endl;

cout <<"Now in order to make your deposit will you please press 1 for savings and 2 for checking "<<endl;

cin >>getac;

if (getac==1) {

cout<<"How much cash would you like to deposit dear customer our machine accepts only solid dollar integers"<<endl;

cin >>dsavings;

savings += dsavings;

cout<<"Your New balance is"<<endl;

cout<<"Savings: " <<savings<<endl;

cout<<"Checking: " <<checking<<endl; }

else if (getac==2) {

cout<<"how much cash mr brown" <<endl;

cin >>dchecking;

checking += dchecking;

cout<<"Your New balance is"<<endl;

cout<<"Checking: " <<checking<<endl;

cout<<"Savings: "<<savings<<endl;

} }

else {

cout<< "Incorrect password"<<endl;

}

}

else {

cout<< "wrong username buddy"<<endl;

}

# Grade Sheet

*Fundamentals of Programming*

*Ms. Vanessa Coote*

*Before submitting the project package, the student should review each of the elements listed below and put a checkmark only in those checkboxes where the designated elements has been reviewed and meets specifications. After completing your document package, number your pages and write the designated page numbers onto the spaces provided on the grading sheet.*

**\_\_\_\_\_ Professionalism (10 points)**

* Following directions
* Neatly assembled 8 ½ by 11
* Cover page
* Page numbers
* Documentation

**\_\_\_\_\_ Source Program Listing and Proper Execution of Program (30 points)**

*It is expected that each student’s program will run correctly*

* Program source code listing matches code on submission and/or backups
* Inclusion of comment lines in source code
* Comments at the beginning of the program including programmer, project name and number, date written, and brief program description.
* Comments at key locations throughout the code
* Descriptive variable names (that follow naming convention)
* Logic is correct
* Logic is clear and easy to follow
* Proper formatting of statements
* Alignment, proper indentation, etc
* Proper use of data types and data conversions

**\_\_\_\_\_ Test Data (5 points)**

* Each test case properly calculated by hand and documented
* Suitable choice of you own test data case

**\_\_\_\_\_ Input Window (10 points)**

* Correct data type for each input section
* Analysis of data type (e.g. int, float, double etc.)
* Appropriate restrictions for each input section
* Data input value shown matches specified test data
* Appropriate display for each input section

**\_\_\_\_\_ Output (15 points)**

* Suitable layout of output (including required fields, easy to read layout, etc.)
* All data cases displayed
* Correct value displayed for each case
* Correct format of fields (e.g. use of integers and not float as appropriate, dollars and cents, display of $, etc)
* Required output format
* Aesthetics (User-friendliness, easy to understand output, alignments, etc)

**\_\_\_\_\_ Documentation (40 points)**

* Analysis of specifications
* Pseudocode
* Flowchart
* Hard copy of program

**\_\_\_\_\_ Fully Functioning Program (30 points)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Possible points = 140

Points Earned =