|  |
| --- |
| Homework-2 |
| Team#: \_\_\_  Team Member-1:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Member’s Contribution (in %) \_\_  Team Member-2:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Member’s Contribution (in %) \_\_  Team Member-3:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Member’s Contribution (in %) \_\_ |
| **Submission**   1. Work on the following requirements. 2. Rename the CPP file and algorithm file to HW2\_Team# (e.g., HW2\_Team1.cpp) 3. Upload your files to Canvas. |

**Problem Statement:** Write a C++ program that reads a customer’s checking account information from a dat file and calculates his/her account balance. In particular, a menu-based application should be developed to perform the following functionalities iteratively until the user requests to quit the program:

1. **Display the account summary:** This option allows the user to enter a 3-digit customer ID (e.g., 007 for the customer name James Bond). The program should read-in an appropriate dat file (e.g., Cust\_007.txt) corresponding to the customer ID and display the account information, including the total account balance on the console. **[10 points]**
2. **Deposit the amount into the account:** This option allows the user to enter a 3-digit customer ID, add the requested amount at the end of the file, and save the file. The option should display the account information on the console, including the total account balance. **[20 points]**
3. **Withdraw the amount from the account:** This option allows the user to enter a 3-digit customer ID, read the customer’s dat file, verify that the requested withdrawal is less than the account balance, and then add the withdrawal amount at the end of the file. The option should display the account information on the console, including the total account balance.

In case if the requested withdrawal exceeds the account balance, the program should display an error message and terminate the transaction. **[25 points]**

1. **Quit the program:** This option terminates the program. **[5 points]**

Four test files are provided to test your program. The dat files are named based on the customer id. For instance, the bank’s first customer is the book’s author, Tony Gaddis, and his customer ID is Cust\_001. The dat file of his checking account is Cust\_001.dat.

**Input validations:** The menu-based option should not allow any options other than 1 to 4 **[10 points]**. The program should not allow users to enter negative numbers **[10 points]**. As long as the user does not enter the correct values, the program should not move forward and keep asking the user to enter the correct values.

**Programming Style:** Make sure the programming style is followed. **[10 points]**

**Programming Comments:** Please make sure the code is well-commented. **[10 points]**

**Note:** Use of ChatGPT or similar generative AI is not allowed.

**Test Cases:**

1. Cust\_001 request to check his balance
2. Cust\_020 request to deposit $1000 to his account
3. Cust\_007 request to withdraw $100 from his account but he has only $10 account balance