STERLING GERARD

FINAL EXAM CUSTOMER ACCOUNT PSEUDO CODE

// INCLUDE THE REQUIRED HEADER FILES

// USE THE STANDARD NAMESPACE

// OFSTREAM TO WRITE TO FILE OUTFILE

// CREATE CLASS “BANKGENERIC”

// DECLARE PUBLIC MEMBERS IN THE CLASS

// DOUBLE BALANCE

// INT DEPOSITS

// INT COUNT

// INT WITHDRAWLS SET TO 0

// DOUBLE ANNUAL INTEREST RATES

// DOUBLE SERVICE CHARGE SET TO 0

// CREATED A CONSTRUCTOR BANKGENERIC()

// VOID DEPOSIT FUNCTION PASSING DOUBLE AMT

// VOID WITHDRAW FUNCTION PASSING DOUBLE AMT

// VOID CAL INTEREST()

// CREATED A VIRTUAL FUNCTION MONTHLY PROC()

// CREATE CLASS “SAVINGS”

// BOOL STATUS TO CHECK THE SAVINGS ACCOUNT IS ACTIVE = 1 OR INACTIVE = 0

//DECLARE THE PUBLIC MEMBERS OF CLASS “SAVINGS”

// VOID WITHDRAW FUNCTION PASSING DOUBLE AMT

// VOID DEPOSIT FUNCTION PASSING DOUBLE AMT

// VOID MONTHLY PROC()

// CREATE CLASS “CHECKING”

//DECLARE THE PUBIC MEMEBERS OF THE CLASS “CHECKING”

// VOID WITHDRAW FUNCTION PASSING DOUBLE AMT

// VOID MONTHLYPROC()

// BANKGENERIC CLASS INHERITED BANKGENERIC() CONTSTRUCTOR

// PROMPT THE USER TO ENTER STARTING BALANCE

// USER ENTERS BALANCE

//PROMPT USER TO ENTER ANNUAL INTEREST RATE

// USER ENTERS INTEREST

//VOID GENERIC INHERIT DEPOSIT FUNCTION

// DEPOSIT FUNCTION PASSES DOUBLE AMT AS PARAMETERS

// BALANCE EQUALS BALANCE + AMT

// INCREMENT DEPOSITS

// VOID GENERIC INHERIT WITHDRAW FUNCTION

//WITHDRAW FUNCTION PASSES DOUBLE AMT AS PARAMETES

// BALANCE EQUALS AMOUNT MINUS BALANCE

// WITHDRAWLS INCREMENTS

// VOID BANKGENERIC INHERIT CALCULATE INTEREST FUNCTION

// CALCULATE THE MONTHLY INTEREST = ANNUAL INTEREST RATE / 12 MONTHS

// BALANCE \* MONTHLY INTEREST RATE = MONTHLY INTEREST RATE

// BALANCE + MONTHLY INTEREST = BALANCE

// VOID BANKGENRIC INHERIT MONTHLY PROC()

// CALL CALCULATE INTERST FUNCTION()

//BALANCE EQUALS BALANCE - SERVICE CHARGES

// WITHDRALS IS ACCUMULATOR SET TO 0

// DEPOSITS IS ACCUMULATOR SET TO 0

// SERVICE CHARGES IS ACCUMULATOR SET TO 0

// VOID SAVINGS INHERIT WITHDRAW FUNCTION

// WITHDRAW FUNCTION PASSES DOUBLE AMT

// IF STATUS EQUALS 0

// PROMPT USER SORRY YOU WILL NOT BE ABLE TO WITHDRAW

// IF STATUS

// BANKGENERIC CLASS INHERIT WITHDRAW FUNCTION TO PASS IN THE AMT

// IF BALANCE IS LESS THAN 25 STATUS EQUALS 0

// INPUT VALIDATION

//VOID SAVINGS INHERIT DEPOSIT FUNCTION

// DEPOSIT FUNCTION PASSES DOUBLE AMT

// IF THE AMT + BAL IS GREATER THAN $25 THEN YOUR STATUS IS 1

// BANKGENERIC INHERIT THE DEPOSIT FUNCTUION SO YOU PASS THE AMT

// INPUT VALIDATION

// VOID SAVINGS INHERIT MONTHLY PROC

// IF WITHDRAWLS ARE GREATER THAN 4

// SERVICE CHARGES EQUALS WITHDRAWLS – 4

// IF BALANCE IS LESS THAN 25

// STATUS = 0

// PROMPT SERVICE CHARGES TO USER

// BANKGENERIC INHERIT MONTHLYPROC()

// INPUT VALIDATION

// VOID CHECKING INHERIT WITHDRAW FUNCTION

// WITHDRAW FUNCTION PASSES DOUBLE AMT

// IF THE USER BALANCE – AMT IS LESS THAN 0

// BALANCE EQUALS BALANCE MINUS 15

// SERVICE CHARGES EQUALS SERVICE CHARGES + 15

// BANKGENERIC INHERIT WITHDRAWLS FUNCTION PASSING AMT AS PARAMETER

// INPUT VALIDATION

//VOID CHECKING INHERIT MONTHLY PROC

// THE SERVICE CHARGE EQUALS 5 + .1 \* WITHDRAWLS

// PROMPT THE USER WITH THE AMOUNT OF SERVICE CHARGES

// BANKGENERIC INHERITS MONTHLYPROC()

// VOID MENU PASSING SAVINGS & OBJECT S, CHECKING REFERENCE OBJECT C

// SELECTION IS SET TO 0

// WHILE SELECTION IS NOT EQUAL TO CHOICE 6

// PROMPT THE USER WITH A BANK ACCOUNT SELECTION MENU

// 1. SAVINGS ACCOUNT DEPOSIT

//2. CHECKING ACCOUNT DEPOSIT

// 3. SAVINGS ACCOUNT DEPOSIT

// 4. CHECKING ACCOUNT DEPOSIT

// 5. UPDATE AND DISPLAY ACCOUNT STATISTICS

// 6. EXIT

// PROMPT USER PLEASE CHOSE BETWEEN CHOICE 1-6

// USER ENTERS A SELECTION

// SWITCH STATEMENT PASSING SELECTION THROUGH THE FUNCTION

//CASE 1

//PROMPT THE USER TO MAKE A SELECTION

//USER ENTERS THE AMT TO DEPOSIT

// PROMPT THE USER THEIR BALANCE

//CASE 2

//PROMPT THE USER TO MAKE A SELECTION

//USER ENTER THE AMT TO DEPOSIT

// PROMPT THE USER THEIR BALANCE

//CASE 3

//PROMPT THE USER TO MAKE A SELECTION

//USER WITHDRAWS

// PROMPT THE USER THEIR BALANCE

//CASE 4

//PROMPT THE USER TO MAKE A SELECTION

//USER WITHDRAWS

// PROMPT THE USER THEIR BALANCE

//CASE 5

// PROMPTS THE USER WITH THE SAVINGS ACCOUNT MONTHLY STATISTICS

// PROMPTS THE USER WITH THE WITHDRAWLS

// PROMPTS THE USER WITH THE DEPOSITS

// THIS IS THE MONTHLY PROC

// PROMPTS THE USER WITH THE ENDING BALANCE

//PROMPTS THE USER TO PRESS A KEY TO CONTINUE

// PROMPTS THE USER THE CHECKING ACCOUNT MONTHLY STATISTICS

//PROMPTS THE USER WITHDRAWLS

// PROMPTS THE USER DEPOSITS

//CALCULATE MONTHLY PROC

//PROMPT THE USER ENDING BALANCE

// PRESS ANY KEY TO CONTINUE

// BREAK STATEMENT

// DEFINE THE MAIN() FUNCTION

//OPEN UP THE OUTFILE OPENS “BANKOUTPUT.TXT”

// SAVINGS CREATED AN OBJECT “S”

// CHECKING CREATED AN OBJECT “C”

//MENU FUNCTION PASSES TWO PARAMETERS (SAVINGS OBJECT S AND CHECKING OBJECT C) CALLING THE FUNCTION

// OUTFILE CLOSES THE FILE THAT WE WERE WRITING TO

// RETURN 0;

// END OF PROGRAM