**C++ Part II (INFO1-CE9265) Spring 2015 – Homework 4**

Clement Chan

**Question 4:**

#include <iostream>

#include <string>

using namespace std;

//The get product function

int getProductID(int ids[], string names[], int numProducts, string target){

for(int i=0; i<numProducts; i++){

if(names[i] == target){

throw ids[i];

}

else

if(names[i] != target && i == numProducts-1){

throw 'N';

}

}

}

int main(){

int productIds[] = {4,5,8,10,13};

string products[] = {"computer", "flash drive", "mouse", "printer", "camera"};

try{

getProductID(productIds, products, 5, "mouse");

}catch(int a){

cout << "The returned product id is: " << a << endl;

}

catch(...){

cout << "There is no match on the input..." << endl;

}

try{

getProductID(productIds, products, 5, "camera");

}catch(int a){

cout << "The returned product id is: " << a << endl;

}

catch(...){

cout << "There is no match on the input..." << endl;

}

try{

getProductID(productIds, products, 5, "laptop");

}catch(int a){

cout << "The returned product id is: " << a << endl;

}

catch(...){

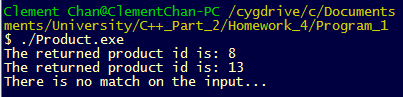
cout << "There is no match on the input..." << endl;

}

return 0;

}

**Output**



**Question 5:**

#include <iostream>

#include <string>

using namespace std;

class Account{

private:

double balance;

public:

//Constructor

Account(){

balance = 0;

}

//Parameterized Constructor

Account(double initialDeposit){

balance = initialDeposit;

}

//Getter

double getBalance(){

return balance;

}

//Inputting deposit function

double deposit(double amount){

if(amount > 0){

balance += amount;

return getBalance();

}else if(amount == 0){

return getBalance();

}else{

throw 'N';

}

}

//Withdrawing deposit function

double withdrawal(double amount){

if(amount > balance || amount < 0){

throw 'N';

}else{

balance -= amount;

return getBalance();

}

}

};

int main(){

Account A1;

int dollar;

char DW;

char ans;

do{

cout << "Please choose if you want to deposit or withdraw: (D-deposit, W-withdraw): " << endl;

cin >> DW;

try{

if(DW == 'D' || DW == 'd'){

try{

cout<< "Please enter the amount you want to deposit in: " << endl;

cin >> dollar;

cout << "The total deposit now is: " << A1.deposit(dollar)<< " dollars

"<<endl;;

}catch(...){

cout << "There is an error in deposit ... Deposit cannot be below zero "

<<endl;

exit(1);

}

}

else if(DW =='W' || DW == 'w'){

try{

cout<< "Please enter the amount you want to withdraw: " << endl;

cin >> dollar;

cout << "The total deposit now is: " << A1.withdrawal(dollar)<< " dollars

"<<endl;;

}catch(...){

cout << "There is an error withdrawal ... Withdrawl cannot be zero or

exceed balance " <<endl;

exit(1);

}

}

else{

throw 0 ;

}

}

catch(int a){

cout << "Please enter a choice in D/W" << endl;

exit(1);

}

cout << "Continue? (Y/N)" << endl;

cin >> ans;

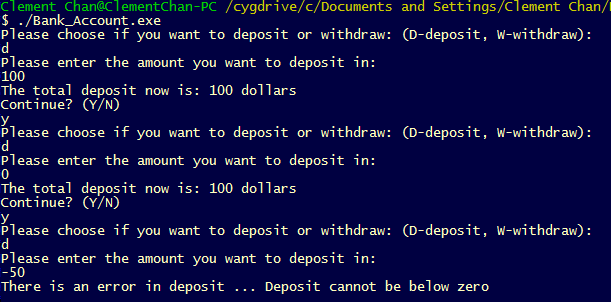
}while(ans == 'Y' || ans == 'y');

return 0;

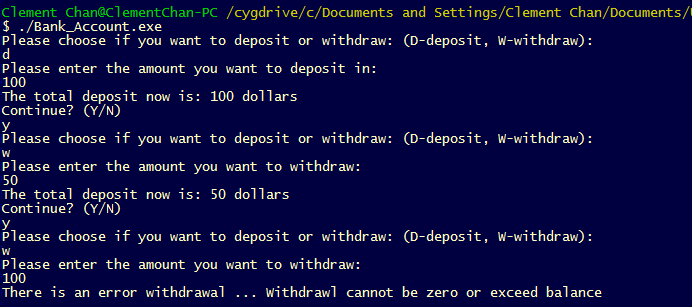
}

**Output**

Test with deposits exceptions



Test with withdrawal exceptions



Test with Choice exceptions

