Shopping Cart Project

Table of Contents

Statement of Independent Effort

I, JaKeyvan Jones hereby certify that is my original work completed without the assistance of anyone or any outside resources.

JaKeyvan Jones

Analysis of Specification

Input

Input username "customer inputs their username for the website"

Input password "customer inputs password for the website"

Input account_num, memberlvl "customer enter account number and member level to authenticate their account"

Input item_name and sku " enters item name and sku number to ensure the correct product pops up"

Input quantity " customer inputs how much of the item they would like to purchase"

Output

Display "Enter user name" "asks customer enters username for website"

Display "Enter pass word" "asks customer enters password for website"

Display " Verify account by entering account number and member level" "message pops up on screen asking customer to verify his/her account"

Display "Access granted" "confirming message that customer has gained access to the website"

Display "Enter product name and stock keeping unit" "asks customer to enter the product of stock keeping unit of the product they're looking for"

Display " How many would you like to purchase?" "requires the customer to see how many items they would like to buy"

Display "Error! Invalid choice! Product name and stock keeping unit do not match" "displays message that product name and sku don't match meaning they entered something wrong into the system"

Display "your total is" "puts total across the sreen"

Display " Error! You do not have enough available funds to purchase items" "error message letting the customer they've surpassed their limit and don't have enough funds to make the purchase"

Display " Thank you for your purchase. Your receipt is below" "thank you message that also lets customers know their receipt is below"

Display "Receipt" "shows receipt and everything important on it"

Display user_name

Display name

Display add_ress

Display user_name

Display item_name, sku

Display price, item_unit, quanitity

Display total

Display account_bal

^^ "all factors that are important for the customer to see on their receipt "

Pseudocode

BEGIN
Declare string name, user_name, add_ress
Declare string productname1 = "1/2 inch bolt", productname2 = "1/4 inch nail", productname3 = "Hammer";
Declare string sku, item_name, item_unit, price, quantity, account_bal;
Declare string Bsku = "HF-342", Nsku = "LK-322", Hsku = "KF-231";
Declare int Bitem_unit = 50, Nitem_unit = 25, Hitem_unit = 1
Declare float Bprice = 20.00 , Nprice = 5.75, Hprice = 15.23;
Declare int Bquantity = 200, Nquantity = 76, Hquantity = 100;
Declare float store_credit = 3000.00
Declare int account_num = 123456789;
Declare real salestax = .06
Display "Enter user name"
Input username
Display "Enter pass word"
Input password
Display " Verify account by entering account number and member level"
Input account_num, memberlvl
Display "Access granted"
Display "Enter product name and stock keeping unit"
Input item_name and sku
IF Item_name = product name and sku = product sku
Display " How many would you like to purchase?"
Input quantity
ELSEIF item_name != product name, sku != product sku
Display "Error! Invalid choice! Product name and stock keeping unit do not match"

Display "your total is"
ASSIGN total = (item_unit * quantity * price) + (salestax)
IF total > account_bal
Display " Error! You do not have enough available funds to purchase items"
ELSEIF total <= account_bal
Display " Thank you for your purchase. Your receipt is below"
Display "Receipt"
Display user_name
Display name
Display add_ress
Display user_name
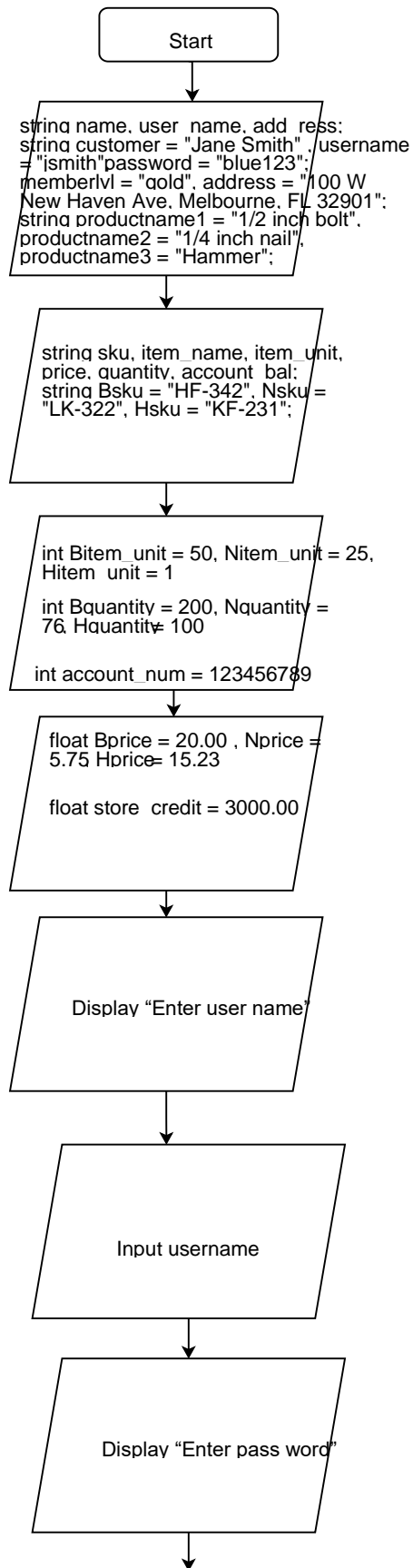Display item_name, sku
Display price, item_unit, quanitity
Display total
Display account_bal

Flowchart

Start

string name, user_name, add_ress;
string customer = "Jane Smith", username
= "jsmith"password = "blue123";
memberlvl = "gold", address = "100 W
New Haven Ave, Melbourne, FL 32901";
string productname1 = "1/2 inch bolt",
productname2 = "1/4 inch nail",
productname3 = "Hammer";

string sku, item_name, item_unit,
price, quantity, account_bal;
string Bsku = "HF-342", Nsku =
"LK-322", Hsku = "KF-231";

int Bitem_unit = 50, Nitem_unit = 25,
Hitem_unit = 1

int Bquantity = 200, Nquantity =
76, Hquantity= 100

int account_num = 123456789

float Bprice = 20.00 , Nprice =
5.75 Hprice= 15.23

float store_credit = 3000.00

Display "Enter user name"

Input username

Display "Enter pass word"

Display " Verify account by entering account number and member level"

Input account_num, memberlvl

Display "Access granted"
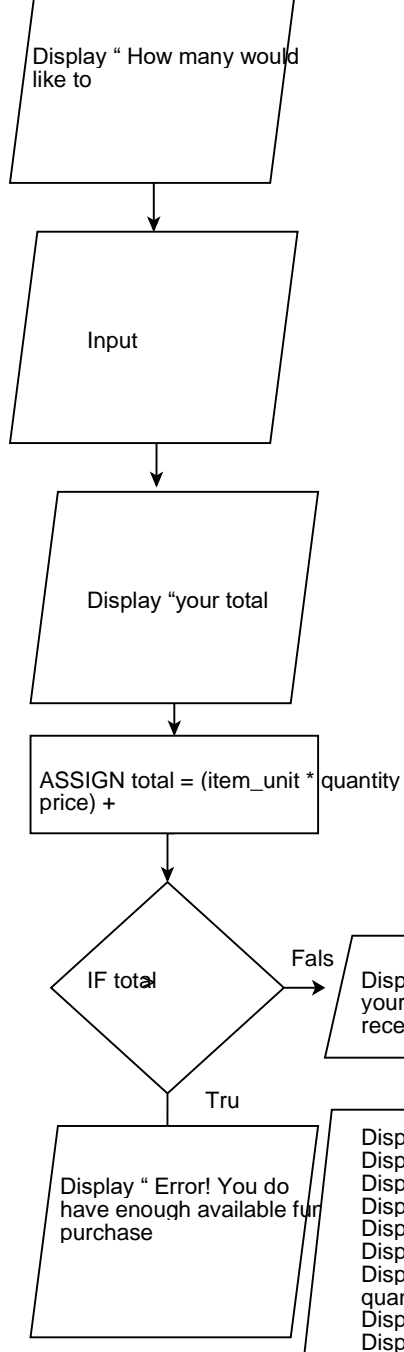
Display "Enter product name and stock keeping unit"

Input item_name and sku

IF Item_name = product name and sku = product sku

False

Display "Error! Invalid choice! Product name and stock keeping unit do not match"

True

Display " How many would like to

Input

Display "your total

ASSIGN total = (item_unit * quantity price) +

IF total

Fals → Display " Thank you your purchase. receipt is

Tru

Display " Error! You do have enough available fur purchase

Display
Display
Display name
Display
Display
Display item_name,
Display price,
quanitity
Display total
Display

Test Cases

SKU	Name	Items in Unit	Price Per Unit	Quantity Purchased	Total	Result	Account Bal		
HF-342	½ inch bolt	50	20.00	2	$2120.00	Pass	$880		
LK-322	¼ inch nail	25	5.75	7	$1066.63	Pass	$1933.37		
HF-342	½ inch bolt	50	20.00	3	$3180.00	Fail, inefficient funds			
KF-231	Hammer	1	15.23	1000	$1614.38	Pass	$1385.62		
HF-342	Hammer	1	15.23			Fail, product name and sku do not match			

Code

```cpp
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;
int main()
{

string name, user_name, add_ress;
string customer = "Jane Smith" , username = "jsmith",  password = "blue123"; memberlvl =
"gold", address = "100 W New Haven Ave, Melbourne, FL 32901"; // declare variables for
customer information
string productname1 = "1/2 inch bolt",  productname2 = "1/4 inch nail", productname3 =
"Hammer"; // declare variables for product names
string sku, item_name, item_unit, price, quantity, account_bal; // declare variables for item
purchases
string Bsku = "HF-342", Nsku = "LK-322", Hsku = "KF-231"; // declare variables for stock
keeping unit name for each item
int Bitem_unit = 50, Nitem_unit = 25, Hitem_unit = 1; // declare variables for number of units
within item
float Bprice = 20.00 , Nprice = 5.75, Hprice = 15.23; // declare variables for price of each item
int Bquantity = 200, Nquantity = 76, Hquantity = 100; // declare varibles for quanitity of each
item
float store_credit = 3000.00; // declare variable for store credit
int account_num = 123456789;  // declare variable for account number
float salestax = .06; // declare variable for sales tax
cout << "Enter user name" << endl; // tells customer to enter user name
cin >> username; // inputs user name
cout << "Enter pass word" << endl; // tells customer to enter pass word
cin >> password; // inputs pass word
cout << "Verify account by entering account number and member level" << endl; // tells
customer to authenitcate accout by entering account number and member level
cin >> account_num, memberlvl; //inputs account number and member level
cout << "Access granted" << endl; // authenitcates username and password
if item_name = product name, sku = product sku {
```

cout << "Enter product name and stock keeping unit" << endl; // tells customer to enter product name and sku

elseif item_name != product name, sku != product sku {

cout << "Error! Invalid choice! Product name and stock keeping unit do not match" << endl; // displays error message because product name or sku does not match

cin >> item_name, sku; // inputs item name and sku

cout << "How many would you like to purchase?" << endl;// asks customer the quantity they would like to purchase

cin >> quantity; // inputs quantity

cout << "Your total is" << endl; // displays total

total = (item_unit * quanitity * price)(salestax) // calculation for total

if (total > account_bal) {

cout << "Error! You do not have enough available funds to purchase items" << endl; // displays error message of you do not have enough money to purchase items

elseif (total <= account_bal) {

cout << "Thank you for your purchase. Your receipt is below" << endl; // displays thank you message and receipt when you purchase goes through

cout << "Receipt" << endl; // display receipt

cout << user_name << endl; // display user name

cout << name << endl; // display name

cout << add_ress << endl; // display address

cout << user_name << endl; // display username

cout << item_name, sku << endl; // display item name and stock keeping unit

cout << price, item_unit, quanitity << endl; // display price, item unit, and quantity

cout << total << endl; // display total

cout << account_bal << endl; // display account balance

account_bal = store_credit - total // displays final account balance

Case1

cout << "Enter Product name and stock keeping unit" << endl;

cin >> productname1, Bsku; // inputs 1/2 inch bolt and HF-342

if item_name = 1/2 inch bolt , sku = HF-342 {

cout << "How many would you like to purchase?" << endl;

cin >> quantity // inputs how many items you purchase

quanitity = 2 // assign quanitity value to 2

cout << "Your total is $2120.00" << endl;

total = (Bitem_unit * 2 * Bprice) + (salestax) // calculates total

cout << "Thank you for your purchase. Your receipt is below" << endl; // displays thank you messsage and receipt

cout << username << endl;

```
cout << name << endl;
cout << address << endl;
cout << item_name << endl;
cout << price, item_unit, quantity << endl;
cout total << account bal << endl;
account_bal = store_credit - total; // assigns equation to total
```

Case2
```
cout << "Enter Product name and stock keeping unit" << endl;
cin >> productname2, Nsku;  // inputs 1/4 inch nail and LK-322
if item_name = 1/4 inch nail, sku = LK-322;
cout << "How many would you like to purchase?" << endl;
cin >> quantity
quantity = 7 // assign quantity value to 7
cout << "Your total is $1066.63" << endl;
total = (Nitem_unit * 7 * Nprice) + (salestax);
cout << "Thank you for your purchase. Your receipt is below" << endl;
cout << username << endl;
cout << name << endl;
cout << address << endl;
cout << item_name << endl;
cout << price, item_unit, quantity << endl;
cout total << account bal << endl;
```

Case3
```
cout << "Enter Product name and stock keeping unit" << endl;
cin  >> productname1, Bsku; // inputs 1/2 inch bold and HF-342
if item_name = 1/2 inch nail, sku = HF-342 {
cout << "How many would you like to purchase?" << endl;
cin >> quantity;
quantity = 3; // assign quantity to 3
cout << "Your total is $3180.00 " << endl;
elseif (total > account_bal)
cout << "Error! You do not have enough available funds to purchase items" << endl; // displays
error because the total is more than the account balance
```

Case4
```
cout << "Enter Product name and stock keeping unit" << endl;
cin >> productname3, Hsku; // inputs Hammer and KF-231
if item_name = Hammer, sku = KF-231 {
cout << "How many would you like to purchase?" << endl;
```

cin >> quantity;
quantity = 100; // assign quantity to 100
cout << "Your total is $1614.38 " << endl;
total = (Hitem_unit * 100 * Hprice) + (salestax)
cout << "Thank you for your purchase. Your receipt is below" << endl;
cout << username << endl;
cout << name << endl;
cout << address << endl;
cout << item_name << endl;
cout << price, item_unit, quantity << endl;
cout total << account bal << endl;

Case 5
cout << "Enter Product name and stock keeping unit" << endl;
cin >> productname3, Bsku; // inputs Hammer and HF-342
if item_name = Hammer, sku != HF-342 {
cout << "Error! Invalid choice! Product name and stock keeping unit do not match" << endl; //
displays error because product name and stock keeping number do not match

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 =