Adamson University  
College of Engineering  
Computer Engineering Department

Data Structures & Algorithm Analysis Lab  
MW/7am-10am/CL3

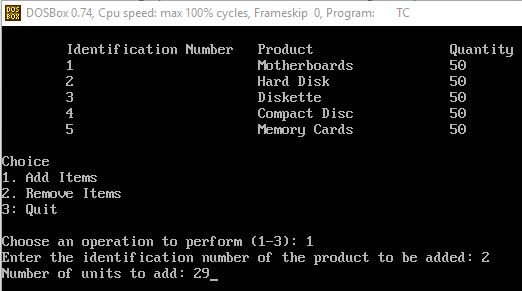
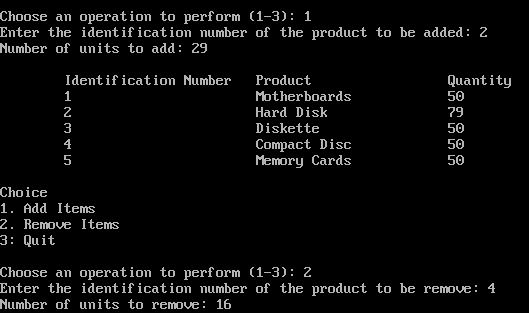
Supplementary Exercise #4  
STRUCTURES

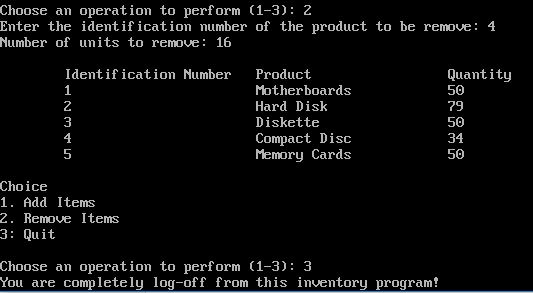
Submitted by:  
Luyon, James Matthew P.

Submitted to:  
Engr. Maria Concepcion A. Mirabueno

March 27, 2019  
Date

Screen Shots: “STRCTSUP”





II. The Source Code

(STRCTSUP)

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

void main()

{

clrscr();

struct product

{

float motherboard[5];

float harddisk[5];

float diskette[5];

float compact\_disc[5];

float memory\_cards[5];

int n,i[1],choice,index;

};

product p;

p.motherboard[p.index]=50;

p.harddisk[p.index]= 50;

p.diskette[p.index]= 50;

p.compact\_disc[p.index]= 50;

p.memory\_cards[p.index]= 50;

cout<<"\n\tIdentification Number\t"<<"Product \t\t"<<"Quantity";

cout<<"\n\t1 \t\t"<<"Motherboards \t\t"<<p.motherboard[p.index];

cout<<"\n\t2 \t\t"<<"Hard Disk \t\t"<<p.harddisk[p.index];

cout<<"\n\t3 \t\t"<<"Diskette \t\t"<<p.diskette[p.index];

cout<<"\n\t4 \t\t"<<"Compact Disc \t\t"<<p.compact\_disc[p.index];

cout<<"\n\t5 \t\t"<<"Memory Cards \t\t"<<p.memory\_cards[p.index];

while(p.choice!=3)

{

cout<<"\n\nChoice";

cout<<"\n1. Add Items";

cout<<"\n2. Remove Items";

cout<<"\n3: Quit";

cout<<"\n\nChoose an operation to perform (1-3): ";

cin>>p.choice;

if(p.choice == 1)

{

cout<<"Enter the identification number of the product to be added: ";

cin>>p.n;

cout<<"Number of units to add: ";

cin>>p.i[0];

if(p.n==1)

{p.motherboard[0]= p.motherboard[0] + p.i[0];}

if(p.n==2)

{p.harddisk[0]= p.harddisk[0] + p.i[0];}

if(p.n==3)

{p.diskette[0]= p.diskette[0] + p.i[0];}

if(p.n==4)

{p.compact\_disc[0]= p.compact\_disc[0] + p.i[0];}

if(p.n==5)

{p.memory\_cards[0]= p.memory\_cards[0] + p.i[0];}

cout<<"\n\tIdentification Number\t"<<"Product \t\t"<<"Quantity";

cout<<"\n\t1 \t\t"<<"Motherboards \t\t"<<p.motherboard[0];

cout<<"\n\t2 \t\t"<<"Hard Disk \t\t"<<p.harddisk[0];

cout<<"\n\t3 \t\t"<<"Diskette \t\t"<<p.diskette[0];

cout<<"\n\t4 \t\t"<<"Compact Disc \t\t"<<p.compact\_disc[0];

cout<<"\n\t5 \t\t"<<"Memory Cards \t\t"<<p.memory\_cards[0];

}

if (p.choice == 2)

{

cout<<"Enter the identification number of the product to be remove: ";

cin>>p.n;

cout<<"Number of units to remove: ";

cin>>p.i[0];

if(p.n==1)

{p.motherboard[0]= p.motherboard[0] - p.i[0];}

if(p.n==2)

{p.harddisk[0]= p.harddisk[0] - p.i[0];}

if(p.n==3)

{p.diskette[0]= p.diskette[0] - p.i[0];}

if(p.n==4)

{p.compact\_disc[0]= p.compact\_disc[0] - p.i[0];}

if(p.n==5)

{p.memory\_cards[0]= p.memory\_cards[0] - p.i[0];}

cout<<"\n\tIdentification Number\t"<<"Product \t\t"<<"Quantity";

cout<<"\n\t1 \t\t"<<"Motherboards \t\t"<<p.motherboard[0];

cout<<"\n\t2 \t\t"<<"Hard Disk \t\t"<<p.harddisk[0];

cout<<"\n\t3 \t\t"<<"Diskette \t\t"<<p.diskette[0];

cout<<"\n\t4 \t\t"<<"Compact Disc \t\t"<<p.compact\_disc[0];

cout<<"\n\t5 \t\t"<<"Memory Cards \t\t"<<p.memory\_cards[0];

}

if(p.choice == 3)

{

cout<<"You are completely log-off from this inventory program!";

getch();

}

if(p.choice > 3)

{cout<<"Invalid choice";}

}

}