1. **Implement data structures in an executable programming language in the context of well-defined problems. (P2.1)**

**System **

* **Develop a working system as specified in Assessment1 (Phase 1) using selected data structure with the proposed**

**Classes**

** Methods**

** Data Structure**

* **Manipulate data with these operations**

** Add**

** Delete**

** Search**

** Sorting**

* **Provide the system with**

** recursive algorithm**

* **Submit softcopy of the system and printed coding**

1. **Implement opportunities for error handling and reporting. (P2.2)**

* **Provide an appropriate system feedback for error handling and provide appropriate reporting if user enter invalid input.**

**Coding of the Xiaomi Shopping Cart system**

**Program**

using System;

using System.Collections.Generic;

using System.Text;

using System.Collections;

namespace XIAOMI\_ASSIGNMENT

{

class Program

{

static void Main(string[] args)

{

method item\_list = new method();

string choice;

Queue queue = new Queue();

while (true)

{

Console.WriteLine();

Console.WriteLine("|-----------------WELCOME TO XIOAMI SHOPPING CART SYSTEM-------------|");

Console.WriteLine("| |");

Console.WriteLine("|-------------------------------MAIN MENU----------------------------|");

Console.WriteLine("| |");

Console.WriteLine("| A- PRODUCT CATEGORY |");

Console.WriteLine("| B- VIEW SHOPPING CART |");

Console.WriteLine("| C- UPDATE QUANTITY OF PRODUCT |");

Console.WriteLine("| D- SEARCH PRODUCT |");

Console.WriteLine("| E- REMOVE PRODUCT FROM SHOPPING CART |");

Console.WriteLine("| F- EXIT |");

Console.WriteLine("| |");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" ENTER YOUR CHOICE: ");

choice = Console.ReadLine();

choice = choice.ToUpper();

switch (choice)

{

case "A":

Console.WriteLine();

item\_list.getdata();

break;

case "D":

Console.WriteLine();

item\_list.findData();

break;

case "B":

Console.WriteLine();

item\_list.Sort();

break;

case "C":

Console.WriteLine();

item\_list.update\_data();

break;

case "E":

Console.WriteLine();

item\_list.deleteData();

break;

case "F":

Environment.Exit(0);

break;

default:

Console.WriteLine();

Console.WriteLine(" INVALID CHOICE. PLEASE ENTER CORRECT CODE!!!");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine();

break;

}

}

}

}

}

**Class Item**

using System;

using System.Collections.Generic;

using System.Text;

using System.Collections;

namespace XIAOMI\_ASSIGNMENT

{

class item

{

public string itemname;

public int itemcode;

public double price;

public int quantity;

public double Subtotal;

public double totalprice;

}

}

**Class Method**

using System;

using System.Collections.Generic;

using System.Text;

using System.Collections;

namespace XIAOMI\_ASSIGNMENT

{

class method

{

protected ArrayList member\_data = new ArrayList();

protected ArrayList items\_data = new ArrayList();

item itemDetail = new item();

public void getdata()

{

string choice = "Y";

do

{

item temp\_data1 = new item();

string categoryCode;

Console.WriteLine();

Console.WriteLine("|-----------------PLEASE SELECT PRODUCTS CATEGORY--------------------|");

Console.WriteLine("| |");

Console.WriteLine("| A - MI PHONES |");

Console.WriteLine("| B - SMART DEVICES |");

Console.WriteLine("| C - AUDIO |");

Console.WriteLine("| |");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" CATEGORY CODE: ");

categoryCode = Console.ReadLine();

categoryCode = categoryCode.ToUpper();

switch (categoryCode)

{

case "A":

Console.WriteLine();

Console.WriteLine("|-----------------------------MI PHONES------------------------------|");

Console.WriteLine("| |");

Console.WriteLine("|-----------------------PLEASE SELECT PRODUCT------------------------|");

Console.WriteLine("| |");

Console.WriteLine("| CODE PRODUCT NAME PRICE |");

Console.WriteLine("| 1000 POCOPHONE RM1,199 |");

Console.WriteLine("| 1001 MI MAX 3 RM1,079 |");

Console.WriteLine("| 1002 MI 8 LITE RM999 |");

Console.WriteLine("| |");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" PRODUCT CODE:");

temp\_data1.itemcode = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("");

if (temp\_data1.itemcode == 1000)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("POCOPHONE");

temp\_data1.price = 1199;

temp\_data1.totalprice = temp\_data1.quantity \* 1199;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 1001)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI MAX 3");

temp\_data1.price = 1079;

temp\_data1.totalprice = temp\_data1.quantity \* 1079;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 1002)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI 8 LITE");

temp\_data1.price = 999;

temp\_data1.totalprice = temp\_data1.quantity \* 999;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else

{

Console.WriteLine(" INVALID CODE");

}

break;

case "B":

Console.WriteLine();

Console.WriteLine("|---------------------------SMART DEVICES----------------------------|");

Console.WriteLine("| |");

Console.WriteLine("|-----------------------PLEASE SELECT PRODUCT------------------------|");

Console.WriteLine("| |");

Console.WriteLine("| CODE PRODUCT NAME PRICE |");

Console.WriteLine("| 2000 MI DASHCAM RM249 |");

Console.WriteLine("| 2001 VR PLAY RM256 |");

Console.WriteLine("| 2002 MI BAND 2 RM82 |");

Console.WriteLine("| 2003 MI CAMERA KIT RM1,099 |");

Console.WriteLine("| |");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" PRODUCT CODE:");

temp\_data1.itemcode = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("");

if (temp\_data1.itemcode == 2000)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI DASHCAM");

temp\_data1.price = 249;

temp\_data1.totalprice = temp\_data1.quantity \* 249;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 2001)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("VR PLAY");

temp\_data1.price = 256;

temp\_data1.totalprice = temp\_data1.quantity \* 256;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 2002)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI BAND 2");

temp\_data1.price = 82;

temp\_data1.totalprice = temp\_data1.quantity \* 82;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 2003)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI CAMERA KIT");

temp\_data1.price = 1099;

temp\_data1.totalprice = temp\_data1.quantity \* 1099;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else

{

Console.WriteLine(" INVALID CODE");

}

break;

case "C":

Console.WriteLine();

Console.WriteLine("|------------------------------AUDIO---------------------------------|");

Console.WriteLine("| |");

Console.WriteLine("|-----------------------PLEASE SELECT PRODUCT------------------------|");

Console.WriteLine("| |");

Console.WriteLine("| CODE PRODUCT NAME PRICE |");

Console.WriteLine("| 3000 MI SPEAKER RM80 |");

Console.WriteLine("| 3001 MI HEADPHONES RM185 |");

Console.WriteLine("| 3002 POCKET SPEAKER RM98 |");

Console.WriteLine("| |");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" PRODUCT CODE: ");

temp\_data1.itemcode = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("");

if (temp\_data1.itemcode == 3000)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI SPEAKER");

temp\_data1.price = 80;

temp\_data1.totalprice = temp\_data1.quantity \* 80;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 3001)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("MI HEADPHONES");

temp\_data1.price = 185;

temp\_data1.totalprice = temp\_data1.quantity \* 185;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else if (temp\_data1.itemcode == 3002)

{

Console.Write(" PLEASE ENTER QUANTITY: ");

temp\_data1.quantity = Convert.ToInt32(Console.ReadLine());

temp\_data1.itemname = ("POCKET SPEAKER");

temp\_data1.price = 98;

temp\_data1.totalprice = temp\_data1.quantity \* 98;

Console.WriteLine("");

Console.WriteLine(" TOTAL PRICE: RM" + temp\_data1.totalprice);

Console.WriteLine("");

temp\_data1.Subtotal = temp\_data1.Subtotal + (temp\_data1.totalprice);

}

else

{

Console.WriteLine(" INVALID CODE");

}

break;

default:

Console.WriteLine();

Console.WriteLine(" INVALID CHOICE. PLEASE ENTER CORRECT CODE!!!");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine();

break;

}

items\_data.Add(temp\_data1);

Console.Write(" CONTINUE SHOPPING? (Y/N): ");

choice = (Console.ReadLine()).ToUpper();

if (choice == "N")

break;

} while (choice == "Y");

}

public void update\_data()

{

int code;

int newquantity;

item temp\_data = new item(); //untuk search data

Console.WriteLine("|------------------------UPDATE ITEM QUANTITY------------------------|");

Console.WriteLine("");

Console.Write(" ITEM CODE: ");

code = Convert.ToInt32(Console.ReadLine());

//read data 1 by 1 from arraylist then compare item code with user input

foreach (object data in items\_data)

{

temp\_data = (item)data;

if (temp\_data.itemcode == code)

{

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine(" ITEM DESCRIPTION: " + temp\_data.itemname);

Console.WriteLine(" PRICE: " + temp\_data.price);

Console.WriteLine(" QUANTITY: " + temp\_data.quantity);

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" NEW QUANTITY: ");

newquantity = Convert.ToInt32(Console.ReadLine());

temp\_data.quantity = newquantity; //replace current quantity

temp\_data.totalprice = newquantity \* temp\_data.price;

Console.WriteLine("");

Console.WriteLine(" QUANTITY HAS BEEN UPDATED");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine("");

break;

}

}

if(temp\_data.itemcode != code)

{

Console.WriteLine("");

Console.WriteLine(" INVALID CODE!!!");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine("");

}

}

public void findData()

{

int code;

item temp\_data = new item(); //untuk search data

Console.WriteLine("|----------------------------SEARCH ITEM-----------------------------|");

Console.WriteLine("");

Console.Write(" ITEM CODE: ");

code = Convert.ToInt32(Console.ReadLine());

//read data 1 by 1 from arraylist then compare item code with user input

foreach (object data in items\_data)

{

temp\_data = (item)data;

if (temp\_data.itemcode == code)

{

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine(" ITEM DESCRIPTION: " + temp\_data.itemname);

Console.WriteLine(" PRICE: " + temp\_data.price);

Console.WriteLine(" QUANTITY: " + temp\_data.quantity);

Console.WriteLine("|--------------------------------------------------------------------|");

break;

}

}

if (temp\_data.itemcode != code)

{

Console.WriteLine("");

Console.WriteLine(" INVALID CODE!!!");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine("");

}

}

public void Sort()

{

//declare new array, size = no.of arraylist data

int[] arr\_qty = new int[items\_data.Count];

item temp\_data = new item();

int i = 0;

double subtotal = 0;

double payment, remaining;

string choice;

foreach (object data in items\_data)

{

temp\_data = (item)data;

arr\_qty[i] = temp\_data.itemcode;

i++;

}

//call method mergesort to sort data in arr\_qty

mergesort(arr\_qty, 0, arr\_qty.Length - 1);

Console.WriteLine("");

Console.WriteLine("|---------------------------SHOPPING CART----------------------------|");

Console.WriteLine(" ");

Console.WriteLine(" NO\t CODE\t DESCRIPTION\t PRICE\t QTY\t TOTAL PRICE(RM)");

for (int j = 0; j < arr\_qty.Length; j++)

{

foreach (object data in items\_data)

{

temp\_data = (item)data;

if (arr\_qty[j] == temp\_data.itemcode)

{

Console.WriteLine(" {0}\t {1}\t {2}\t {3}\t {4}\t {5}", j + 1, temp\_data.itemcode,

temp\_data.itemname, temp\_data.price, temp\_data.quantity, temp\_data.totalprice);

subtotal = subtotal + temp\_data.totalprice;

}

}

temp\_data.Subtotal = subtotal;

}

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine(" SUBTOTAL: RM" + temp\_data.Subtotal);

Console.WriteLine("");

Console.Write(" MAKE A PAYMENT (Y/N): ");

choice = Console.ReadLine().ToUpper();

if (choice == "Y")

{

Console.WriteLine("");

Console.WriteLine("|---------------------------MAKE A PAYMENT---------------------------|");

Console.WriteLine("");

Console.WriteLine(" SUBTOTAL: RM" + temp\_data.Subtotal);

Console.Write(" PAYMENT: RM");

payment = double.Parse(Console.ReadLine());

remaining = payment - temp\_data.Subtotal;

Console.WriteLine(" CHANGE DUE: RM" + remaining);

Console.WriteLine("|--------------------------------------------------------------------|");

Console.ReadKey();

}

}

public void mergeArray(int[] arr, int start, int mid, int end)

{

/\* Create a temporary array for stroing merged array (Length of temp array will be

\* sum of size of both array to be merged)\*/

int[] temp = new int[end - start + 1];

int i = start, j = mid + 1, k = 0;

// Now traverse both array simultaniously and store the smallest element of both to temp array

while (i <= mid && j <= end)

{

if (arr[i] < arr[j])

{

temp[k] = arr[i];

k++;

i++;

}

else

{

temp[k] = arr[j];

k++;

j++;

}

}

// If there is any element remain in first array then add it to temp array

while (i <= mid)

{

temp[k] = arr[i];

k++;

i++;

}

// If any element remain in second array then add it to temp array

while (j <= end)

{

temp[k] = arr[j];

k++;

j++;

}

// Now temp has merged sorted element of both array

// Traverse temp array and store element of temp array to original array

k = 0;

i = start;

while (k < temp.Length && i <= end)

{

arr[i] = temp[k];

i++;

k++;

}

}

// Recursive Merge Procedure

public void mergesort(int[] arr, int start, int end)

{

if (start < end)

{

int mid = (end + start) / 2;

mergesort(arr, start, mid);

mergesort(arr, mid + 1, end);

mergeArray(arr, start, mid, end);

}

}

public void deleteData()

{

int code;

string choice;

item temp\_data = new item(); //untuk search data

Console.WriteLine("|----------------------------DELETE ITEM----------------------------|");

Console.WriteLine("");

Console.Write(" ITEM CODE: ");

code = Convert.ToInt32(Console.ReadLine());

//read data 1 by 1 from arraylist then compare item code with user input

foreach (object data in items\_data)

{

temp\_data = (item)data;

if (temp\_data.itemcode == code)

{

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine(" ITEM DESCRIPTION: " + temp\_data.itemname);

Console.WriteLine(" PRICE: " + temp\_data.price);

Console.WriteLine(" QUANTITY: " + temp\_data.quantity);

Console.WriteLine("|--------------------------------------------------------------------|");

Console.Write(" DELETE ITEM? (Y/N): ");

choice = Console.ReadLine().ToUpper();

if (choice == "Y")

{

items\_data.Remove(temp\_data);

Console.WriteLine("");

Console.WriteLine(" DATA HAVE BEEN REMOVED");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine("");

break;

}

}

}

if (temp\_data.itemcode != code)

{

Console.WriteLine("");

Console.WriteLine(" INVALID CODE!!!");

Console.WriteLine("|--------------------------------------------------------------------|");

Console.WriteLine("");

}

}

}

}

**Test results to enable comparison with expected results. (P2.3)**

* **Produce test documentation (valid and invalid input test) using appropriate format.**