Source Code

Class Customer

```
public class Customer {
private boolean member;        private
String name, phone, address;

//normal constructor       public Customer(boolean
member, String name, String phone,
String address) {    this.member = member;
this.name = name;          this.phone = phone;
this.address = address;

}

//setter method public void setMember(boolean
member) {         this.member = member;
}
//setter for name public void setName(String name) {
this.name = name;
}

//accessor       public
boolean isMember(){
return member;    }
public String getPhone() {
return phone;   }      public
String     getAddress()     {
return address;
}
//toString() method (include method overloading)
@Override      public String toString()
{          return
"\nMembership: " + member +
       "\nCustomer Name: " + name +
       "\nCustomer Phone: " + phone +
       "\nCustomer Address: " + address;
    }
}
```

Class Jersey

```
public abstract class Jersey {
protected char design, colour, size;
protected int quantity;      protected
Customer c;

//normal constructor
Public Jersey(char design, char colour, char size,
int quantity, Customer c) {           this.design
= design;           this.colour = colour;
this.size = size;           this.quantity =
quantity;           this.c = c;
      }

//accessor method
public char getDesign() {
return design;       }
public char getColour()
{              return
colour;    }
public char getSize()
{
return size;    }
public int getQuantity() {
return quantity;    }
public Customer getCustomer() {
return c;      }       public
abstract double
calculatePrice();    //toString
method       public String
toString() {          String
list = "";          list +=
"Design: " + design +
        "\nColour: " + colour +
        "\nSize: " + size +
        "\nQuantity: " + quantity +
"\nCustomer Detail:\n";          list
+= c.toString();          return
list;
    }
}
```

Class ShortSleeves

```
public class ShortSleeve extends Jersey {       private char
collar, embroidered;       public ShortSleeve(char design,
char colour, char size, int
quantity, Customer c, char collar, char embroidered) {
super(design, colour, size, quantity, c);
this.collar = collar;
this.embroidered = embroidered;
    }       public char
getCollar() {
return collar;
}       public char
getEmbroidered() {
return embroidered;    }
@Override
public double calculatePrice()
{                double
totPrice = 0.0;
double price1 = 0.0;             if
(getDesign() == 'A') {
price1 = 60.0;
    }  else if (getDesign() ==
'B') {
price1 = 85.0;
    }
else if (getDesign() == 'C') {
price1 = 100.0;
}
 if (getCustomer().isMember()) {
totPrice = price1 * getQuantity()
* 0.85;
// apply15%   discount for members
} else {                  totPrice
= price1 *getQuantity();
}        return
totPrice;
   }
}
```

Class JerseyApps

```
import java.util.Scanner;   import
java.util.StringTokenizer;   import java.io.*;   public
class JerseyApps {           public static void
main(String args[]) throws IOException
    {
    Scanner inputText = new Scanner(System.in);
      boolean continueProgram =
true;
    // Prompt user to start the program
System.out.println("Start the programme? (Y/N): ");
String choice = inputText.next();
if (choice.equalsIgnoreCase("N")) {
continueProgram = false;
}

do {
try {
      int num = 3;
      Customer c[] = new Customer[num];
      Jersey j[] = new Jersey[num];
      // Open custBuy.txt for writing new data
      FileWriter fwCust = new
FileWriter("\"C:\\Users\\abdul\\Desktop\\custBuy.txt",
true);
      BufferedWriter bwCust = new
               BufferedWriter(fwCust);
      PrintWriter pwCust = new PrintWriter(bwCust);
Scanner inputNum = new Scanner(System.in);
int count = 0;
       // Open custBuy.txt for reading existing data
       FileReader fr = new
FileReader"C:\\Users\\abdul\\Desktop\\custBuy.txt");
       BufferedReader br = new BufferedReader(fr);
       String dataRow = br.readLine();
while (dataRow != null) {
            StringTokenizer st = new
                   StringTokenizer(dataRow, "*");
if (st.countTokens() == 10) {  boolean member =

Boolean.parseBoolean(st.nextToken());

String name = st.nextToken();

String phone = st.nextToken();
String address = st.nextToken(); char
design = st.nextToken().charAt(0);
char colour = st.nextToken().charAt(0);
char size = st.nextToken().charAt(0);
int quantity =

Integer.parseInt(st.nextToken());
char collar = st.nextToken().charAt(0);
char embroidered =
st.nextToken().charAt(0);
//store into array of object using normal constructor(the index is represent
by count)
c[count] = new Customer(member, name,
phone, address);
j[count] = new ShortSleeve(design,
colour, size, quantity, c[count], collar, embroidered);
count++;//update the count variable
} else {
    System.out.println("Invalid data format:" + dataRow);    }
//read the new line of data                      dataRow
= br.readLine();
}                  br.close();
// Prompt user for new data   for (int i
= count; i < j.length; i++) {
System.out.println("+----------------------------+");
 System.out.println("| DESIGN |");
System.out.println("+----------------------------+");
 System.out.println("| A | CHOII |");
System.out.println("+----------------------------+");
 System.out.println("| B | SVG |");
System.out.println("+----------------------------+");
System.out.println("| C | HOMEBOIS |");
System.out.println("+----------------------------+");
System.out.println("Design [A|B|C]: ");  char design
= inputText.next().charAt(0);
 System.out.println("+----------------------------+");
 System.out.println("| COLOUR |");
System.out.println("+----------------------------+");
 System.out.println("| G | GREEN |");
System.out.println("+----------------------------+");
System.out.println("| R | RED |");
System.out.println("+----------------------------+");
System.out.println("| Y | YELLOW |");
System.out.println("+----------------------------+");
System.out.println("Colour [G|R|Y]: ");  char colour
= inputText.next().charAt(0);
System.out.println("Size [S|M|L]: ");  char size =
inputText.next().charAt(0);
System.out.println("Quantity: "); int  quantity =
inputNum.nextInt();
```

```
 System.out.println("+----------------------------+");
System.out.println("| COLLAR-TYPE |");
 System.out.println("+----------------------------+");
 System.out.println("| R | RETRO |");
System.out.println("+----------------------------+");
System.out.println("| C | CLASSIC |");
System.out.println("+----------------------------+");
System.out.println("Collar [R|C]: ");  char collar =
inputText.next().charAt(0);
 System.out.println("+---------------------------+");
 System.out.println("| EMBROIDERED-PART |");
 System.out.println("+----------------------------+");
 System.out.println("| F | FULL |");
System.out.println("+----------------------------+");
System.out.println("| N | NAME ONLY |");
System.out.println("+----------------------------+");
System.out.println("Embroidered [F|N]: ");  char
embroidered = inputText.next().charAt(0);
System.out.println("Membership [true|false]:");
boolean member = inputText.nextBoolean();
inputText.nextLine();  // Consume the newline
character  System.out.println("Name: ");
String name = inputText.nextLine();
System.out.println("Phone: ");
String phone = inputText.nextLine();
System.out.println("Address (state): ");
String address = inputText.nextLine();
System.out.println("\nConfirmation:-\n");  c[i]
= new Customer(member, name, phone,  address);
j[i] = new ShortSleeve(design,  colour, size,
quantity, c[i], collar,  embroidered);
pwCust.println("\n" + member +
"" + name + "" + phone + "*" +
address + "*" + design  + "" +
colour + "" + size + "*" +
quantity + "" + collar + ""
+ embroidered);
}  pwCust.close();
bwCust.close();
// Open member.txt for writing customer details
FileWriter fw = new
FileWriter("\"C:\\Users\\abdul\\Desktop\\member.txt";
BufferedWriter bw = new BufferedWriter(fw);
PrintWriter pw = new PrintWriter(bw);
pw.println("\nDetails of customers who have
membership : "); for (int i = 0; i < count; i++)
{       if (c[i].isMember()) {
          pw.println(c[i].toString());
    }
}
```

```
//close all the output file pw.close(); bw.close();
double totalShortSleevePrice = 0;                  for
(int i = 0; i < count; i++) {
totalShortSleevePrice += j[i].calculatePrice();
}
System.out.println("\nThe Grand Total Price: RM"
              + totalShortSleevePrice);
// Open statePortugal.txt for writing customer details
FileWriter fwr = new
FileWriter("\"C:\\Users\\abdul\\Desktop\\statePortugal.txt");
BufferedWriter bwr = new BufferedWriter(fwr);
PrintWriter pwr = new PrintWriter(bwr);
pwr.println("\nDetails of customers from Portugal:");
for (int i = 0; i < count; i++) {                            if
    (c[i].getAddress().equalsIgnoreCase("Portugal")) {
pwr.println(c[i].toString());
      }
}
//close all the output file
pwr.close();
bwr.close();                  int
retroCollarCount = 0;
int classicCollarCount = 0;
for (int i = 0; i < count; i++) {
if (j[i] instanceof ShortSleeve) {
ShortSleeve shortSleeve = (ShortSleeve)
j[i];                               if
(shortSleeve.getCollar() == 'R') {
retroCollarCount++;
      } else if (shortSleeve.getCollar() == 'C') {
classicCollarCount++;
      }
   }
}
System.out.println("Best-selling Jersey Collar Type:");
if (retroCollarCount > classicCollarCount) {
    System.out.println("Retro Collar");
} else if (classicCollarCount > retroCollarCount)   {
System.out.println("Classic Collar");
} else {
    System.out.println("Both Retro Collar and ClassicCollar have the
same number of sales.");
}
System.out.println("\nContinue? (Y/N): ");
choice = inputText.next();                 if
(choice.equalsIgnoreCase("N")) {
continueProgram = false;
 }
} catch (FileNotFoundException e) {
    System.out.println("File not found: " + e.getMessage());
continueProgram = false;
```

```
} catch (IOException e) {
    System.out.println("Error reading or writing file: " +
            e.getMessage());
continueProgram = false;
  }
 } while (continueProgram);
 System.out.println("End of the program.");
}
}
```