# CMPP 239

Assignment Four – Custom Problem

Martin Czerwinski and Nathan Faucher

Dec. 1, 2016

# Problem

To calculate discounts on a large order of computer hardware items. The discounts include a bulk discount and user entered promo codes. The program will display information about the customer’s order and savings.

# Analysis

**Functionality**

Prompt and get the infile name and the outfile name.

Determine the number of rows in the input file.

Extract data from the comma delimited input file lines and parse the input to the appropriate data type.

Tests to see if there are any zero values in the price column, if there are the system will exit.

Transliterates the item names to title case.

Calculates the cost of each item total before discounts are applied.

Applies bulk discount based on quantity of each item.

Prompts the user to enter promo code.

Applies custom promo code discounts to particular item prices in the price array.

Calculates the discounted cost of individual item totals.

Prints and Displays subtotals and totals in a table.

Calculates the highest and lowest savings per item, and the average savings per item.

Prints and Displays highest, lowest and average savings as a table.

**Inputs**

The infile name

The outfile name

User enters promo code names

code (String)

**Promo codes:**

FAMILY price \* 0.95

SPRING15 price \* 0.85

SUMMER15 price \* 0.85

WINTER15 price \* 0.85

FALL15 price \* 0.85

F3qytr Laptop price \* 0.85

Jxn4ns Keyboard price \* 0.75

Mouse price \* 0.75

udXyj6 Switch price \* 0.80

Router price \* 0.80

Gc47wX Ethernet cables \* 0.50

kc42Qj Desktop \* 0.90

Monitor \* 0.70

Keyboard \* 0.70

Mouse \* 0.70

DQB6HS Wireless \* 0.85

Ethernet cable \* 0.85

**Outputs**

itemName (String)

itemPrice (real 2 dp)

itemTotal (real 2 dp)

itemBulk (real 2 dp)

itemDiscount (real 2 dp)

itemQuantity (int)

itemSavings (real 2 dp)

highSavings (real 2 dp)

lowSavings (real 2 dp)

lowIndex (int)

highIndex (int)

avgSavings (real 2 dp)

**Formulas:**

totalPrice[i] = price[i] \* quantity[i]

totalPrice[index] = (price[index] \* specificDiscount) \* quantity[index]

itemSavings[i] = itemTotal[i] - itemDiscount[i]

totalSavings = totalSavings + itemSavings[i]

avgSavings = totalSavings / arraySize

# Test Data

**Input file**

laptop,673.60,15

keyboard,45.60,40

monitor,230.45,30

router,310.81,9

desktop comp,492.90,32

switch,168.46,16

ethernet cable,90.40,18

mouse,13.80,38

wireless router,80.59,9

**Promo code entered:**

Jxn4ns 25% off Mice and Keyboards

**Outputs**

ITEM NAME ITEM QUANTITY ORIGINAL TOTALS DISCOUNT TOTALS

Laptop 15 $10104.00 $ 9598.80

Keyboard 40 $ 1824.00 $ 1231.20

Monitor 30 $ 6913.50 $ 6222.15

Router 9 $ 2797.29 $ 2797.29

Desktop Comp 32 $15772.80 $14195.52

Switch 16 $ 2695.36 $ 2560.59

Ethernet Cable 18 $ 1627.20 $ 1545.84

Mouse 38 $ 524.40 $ 353.97

Wireless Router 9 $ 725.31 $ 725.31

Totals $42983.86 $39230.67

Today You Saved: $ 3753.19

Average Savings per Item: $ 417.02

Number of Records: 9

Highest Savings per Item 32 Desktop Comp $1577.28

Lowest Savings per Item 9 Router $0.00

# Algorithms

**main**

|  |
| --- |
| Prompt and get infile path  Prompt and get outfile path  arraySize = call getArraySize and pass inFilename  while(i<arraySize)  String line= next line  String[] nextfield = line split on ,    itemName[i] = nextfield[0]  itemPrice[i] = Double.parseDouble(nextfield[1])  itemQuantity[i] = Integer.parseInt(nextfield[2])  update i for (i=0; i<arraySize; i++)call checkZer0 to check for zeroitemName[i] = call uppercase and pass itemName[i]itemBulk[i] = call discount.bulk and pass itemQuantity[i] & itemPrice[i]itemTotal[i] = call getTotal and pass itemPrice[i] & itemQuantity[i]while( !code equal "")Prompt and get code inputCall discount.promo and pass codeitemDiscount = call discount.total and pass itemName, itemBulk & itemQuantity call printTitles for (i=0; i<arraySize; i++)call printReceipt and pass itemName[i], itemQuantity[i], itemTotal[i] & itemDiscount[i]for (i=0; i<arraySize; i++)itemSavings[i] = itemTotal[i] - itemDiscount[i]if (itemSavings[i] > highSavings)highSavings = itemSavings[i]highIndex = iif (itemSavings[i] < lowSavings)lowSavings = itemSavings[i]lowIndex = idouble avgSavings = call calcAvgSavings and pass itemTotal, itemDiscount & arraySizecall printTotal and pass itemTotal, itemDiscount, arraySize & avgSavingscall printCalcs and pass itemQuantity[highIndex], itemQuantity[lowIndex], itemName[highIndex], itemName[lowIndex], highSavings & lowSavings |

# **getInFile**

|  |
| --- |
| String filename = prompt and get infile nameFile file = new File(filename)inputFile = new Scanner(file)return filename |

# **getOutFile**

|  |
| --- |
| String Out = prompt and get infile nameoutfile = new UtilityClass(Out);outfile.openFile();return Out; |

# **getTotal**(double itemPrice, int itemQuantity)

|  |
| --- |
| double itemTotal = itemPrice\*itemQuantityreturn itemTotal; |

# **upperCaser**(String itemName)

|  |
| --- |
| String temp1 = itemName.toUpperCase().substring(0,1)itemName = temp1 + itemName.substring(1)int whitespace = indexOf “ “if (whitespace != -1)String temp2 = itemName.toUpperCase().substring(whitespace+1,whitespace+2)itemName = itemName.substring(0,whitespace+1) + temp2 + itemName.substring(whitespace+2)return itemName |

# **checkZer0**(double price, int i)

|  |
| --- |
| if (price == zhero)Print “Error: the price in line %d is zero."Write “Error: the price in line %d is zero."Exit application |

# **getArraySize**(String inFilename)

|  |
| --- |
| int i = 0Open file inFilenameString line = next linewhile(line!=null)i++;tryline=inputFile.nextLine();catch (Exception e)break;return i |

# **calcAvgSavings**(double[] itemTotal, double[] itemDiscount, int arraySize)

|  |
| --- |
| Double[] itemSavings = new Double[arraySize]double totalSavings=0for (int i=0; i<arraySize; i++)itemSavings[i] = itemTotal[i] - itemDiscount[i]totalSavings = totalSavings + itemSavings[i]double avgSavings = totalSavings / arraySizereturn avgSavings |

# **printTitles**()

|  |
| --- |
| Print table headers to console Write table headers to file |

# **printReceipt**(String itemName, int itemQuantity, double itemTotal, double itemDiscount)

|  |
| --- |
| Print itemName, itemQuantity, itemTotal & itemDiscount in a table format to consoleWrite itemName, itemQuantity, itemTotal & itemDiscount in a table format to file |

# **printReceipt**(String itemName, int itemQuantity, double itemTotal, double itemDiscount)

|  |
| --- |
| Print “Total.........................................................."Write “Total.........................................................."for(int i = 0; i < itemTotal.length; i++)tempTotal = tempTotal + itemTotal[i]tempDiscount = tempDiscount + itemDiscount[i]Print tempTotal & tempDiscountWrite tempTotal & tempDiscounttempTotal = tempTotal - tempDiscountPrint tempTotal with “Today You Saved”Write tempTotal with “Today You Saved”Print arraySize as number of records Write arraySize as number of records  Print avgSavings  Write avgSavings |

# **printCalcs**(int highItemQuantity, int lowItemQuantity, String highItemName, String lowItemName, double highSavings, double lowSavings)

|  |
| --- |
| Print highItemQuantity & highItemNameWrite highItemQuantity & highItemNamePrint lowItemQuantity & lowItemNameWrite lowItemQuantity & lowItemName |

**Class Discount**

# **discount.bulk**(int highItemQuantity, int lowItemQuantity, String highItemName, String lowItemName, double highSavings, double lowSavings)

|  |
| --- |
| double priceDisc=0  if (quantity >= 100)  priceDisc = price\*0.80  else if (quantity >= 50)  priceDisc = price\*0.85  else if (quantity >= 25)  priceDisc = price\*0.90  else if (quantity >= 10)  priceDisc = price\*0.95  else  priceDisc = price  return priceDisc |

# **discount.promo**(String Code)

|  |
| --- |
| Boolean flagSwitchActivated = false  int month = 0  Date date = new Date()  month = Integer.parseInt(monthFormat.format(date))  switch(code)  case "":  flagSwitchActivated = true  case "F3qytr":  if (flagCheckLaptop == false)  Print You entered the laptop code  flagSwitchActivated = true  flagCheckLaptop = true  else  Print You already entered the laptop code  case "Jxn4ns":  if (flagCheckComp == false)  Print You entered the computer peripherals code  flagSwitchActivated = true;  flagCheckComp = true;  else  Print You already entered the computer peripherals code  flagSwitchActivated = true  case "udXyj6":  if (flagCheckNetwork == false)  Print You entered the networking code  flagSwitchActivated = true  flagCheckNetwork = true  else  Print You already entered the networking code  flagSwitchActivated = true  case "Gc47wX":  if (flagCheckCable == false)  Print You entered the Ethernet code  flagSwitchActivated = true  flagCheckCable = true  else  Print You already entered the Ethernet code  flagSwitchActivated = true  case "kc42Qj":  if (flagCheckOffice == false)  Print You entered the home office starter pack code  flagCheckOffice = true  flagSwitchActivated = true  else  Print You already entered the home office starter pack code  flagSwitchActivated = true    case "DQB6HS":  if (flagCheckHomeNet == false)  Print You entered the home networking starter pack code  flagCheckHomeNet = true  flagSwitchActivated = true  else  Print You have already entered the home office starter pack  flagSwitchActivated = true  default:  break  if(flagSwitchActivated==false) //if the first switch was not activated  switch(code.toUpperCase())  case "FULLCOMMIE":  Print Glory to the motherland!  case "SPRING15":  if(month==3 || month==4 || month==5)  if(flagCheckSeason == false)  Print You entered the spring 15% off code  flagCheckSeason = true  else  Print You already entered the spring 15% off code  else  Print Invalid option  case "SUMMER15":  if(month==6 || month==7 || month==8)  if(flagCheckSeason == false)  Print You entered the summer 15% off code  flagCheckSeason =true  else  Print You already entered the summer 15% off code  else  Print Invalid option  case "FALL15":  if(month==9 || month==10 || month==11)  if(flagCheckSeason == false)  Print You entered the fall 15% off code  else  Print You already entered the fall 15% off code  else  Print Invalid option  case "WINTER15":  if(month==12 || month==1 || month==2)  if(flagCheckSeason == false)  Print You entered the winter 15% off code  flagCheckSeason=true  else  Print You already entered the winter 15% off code else  Print Invalid option  case "FAMILY":  if (flagCheckFamily == false)  Print You entered the friends and family discount  flagCheckFamily = true  else  Print You already entered the friends and family discount  default:  Print Invalid option  flagSwitchActivated = false |

**discount.total**(String[] names, double[] price, int[] quantity)

|  |
| --- |
| int index = 0;  double[] totalPrice = new double[price.length]  for(int i = 0; i < totalPrice.length; i++)  totalPrice[i] = price[i] \* quantity[i]  if(flagCheckLaptop == true)  index = call ary.getIndex and pass names & "Laptop"  if (index != -1)  totalPrice[index] = (price[index] \* 0.85) \* quantity[index]  else  Print There are no Laptops in your order.  if(flagCheckComp == true)  index = call ary.getIndex and pass names & "Mouse"  if (index != -1)  totalPrice[index] = (price[index] \* 0.75) \* quantity[index]  else  Print There are no Mice in your order.  index = call ary.getIndex and pass names & "Keyboard"  if (index != -1)  totalPrice[index] = (price[index] \* 0.75) \* quantity[index]  else  Print There are no Keyboards in your order.  if(flagCheckNetwork == true)  index = call ary.getIndex and pass names & "Switch"  if (index != -1)  totalPrice[index] = (price[index] \* 0.80) \* quantity[index]  else  Print There are no Switches in your order.  index = call ary.getIndex and pass names & "Router"  if (index != -1)  totalPrice[index] = (price[index] \* 0.80) \* quantity[index];  else  Print There are no Routers in your order.    if(flagCheckCable == true)  index = call ary.getIndex and pass names & "Ethernet Cable"  if (index != -1)  totalPrice[index] = (price[index] \* 0.50) \* quantity[index]  else  Print There are no Ethernet Cables in your order.  if(flagCheckOffice == true)  index = call ary.getIndex and pass names & "Desktop Comp"  if (index != -1)  totalPrice[index] = (price[index] \* 0.90) \* quantity[index]  else  Print There are no Desktop Computers in your order.  index = call ary.getIndex and pass names & "Monitor"  if (index != -1)  totalPrice[index] = (price[index] \* 0.70) \* quantity[index]  else  Print There are no Monitors in your order.  index = call ary.getIndex and pass names & "Mouse"  if (index != -1)  totalPrice[index] = (price[index] \* 0.70) \* quantity[index]  else  Print There are no Mice in your order.  index = call ary.getIndex and pass names & "Keyboard"  if (index != -1)  totalPrice[index] = (price[index] \* 0.70) \* quantity[index]  else  Print There are no Keyboards in your order.  if(flagCheckHomeNet == true)  index = call ary.getIndex and pass names & "Ethernet Cable"  if (index != -1)  totalPrice[index] = (price[index] \* 0.70) \* quantity[index]  else  Print There are no Ethernet Cables in your order.  index = call ary.getIndex and pass names & "Wireless Router"  if (index != -1)  totalPrice[index] = (price[index] \* 0.70) \* quantity[index]  else  Print There are no Wireless Routers in your order.  if(flagCheckSeason == true)  for(int i = 0; i < totalPrice.length; i++)  totalPrice[i] = (price[i]\*0.85) \* quantity[i]  if(flagCheckFamily == true)  for(int i = 0; i < totalPrice.length; i++)  totalPrice[i] = (price[i]\*0.95) \* quantity[i]  return totalPrice |

**Class ArrayUtil**

**ary.getIndex**(String[] array, String search)

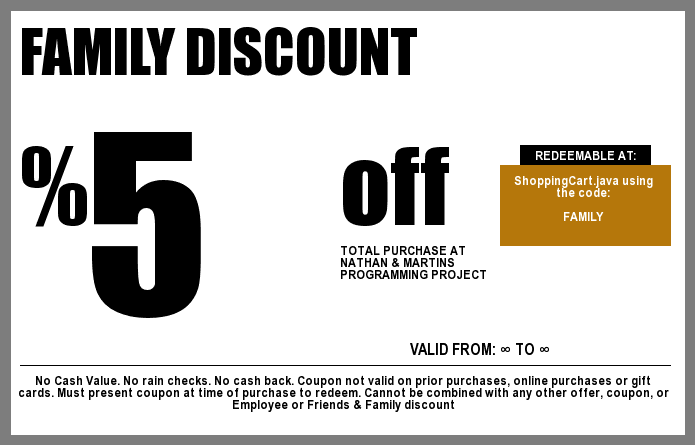
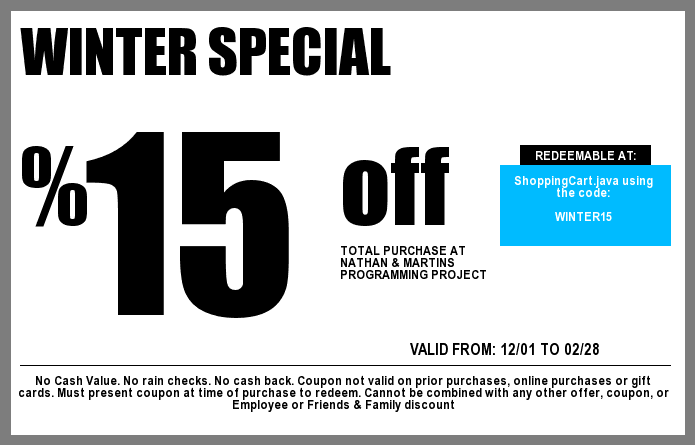
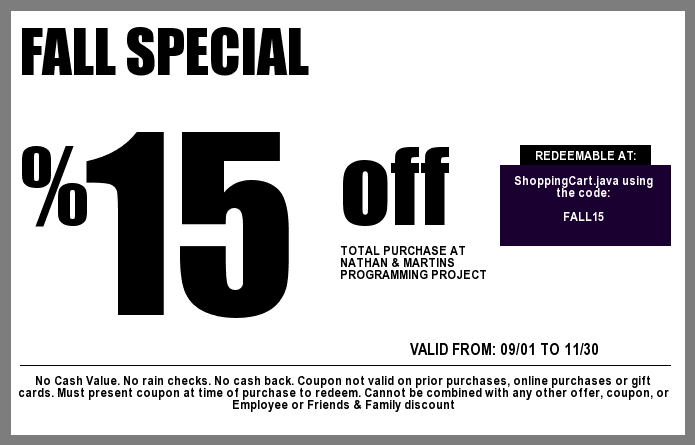
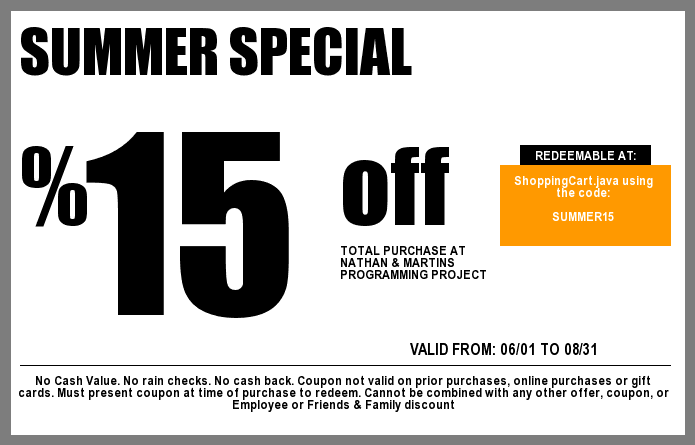
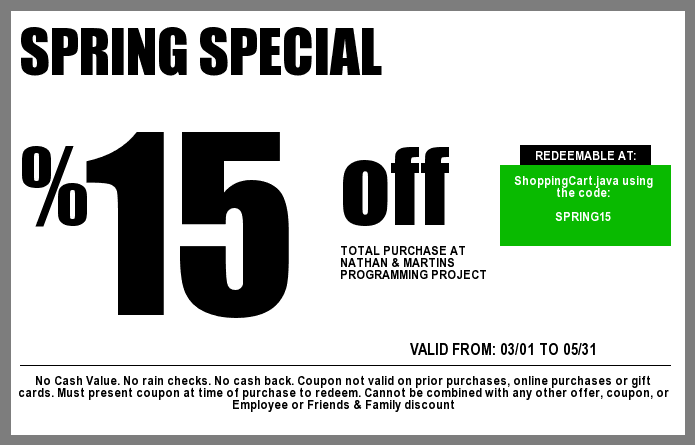
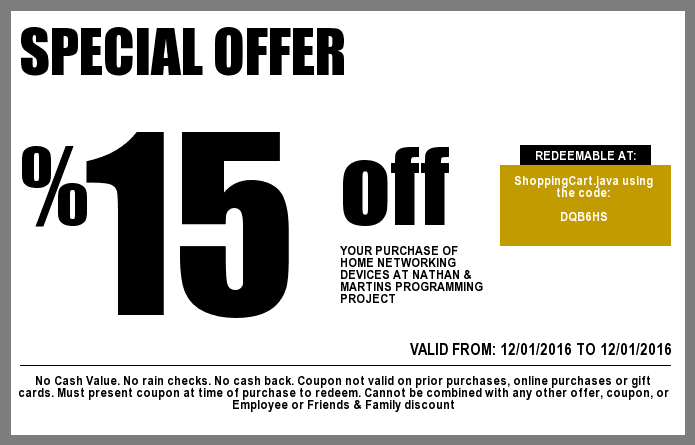
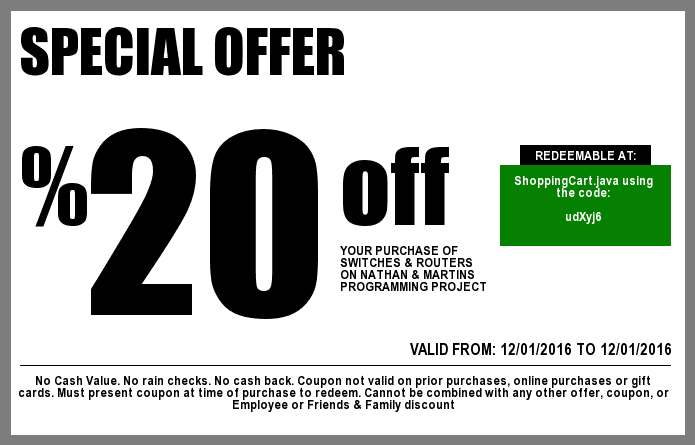
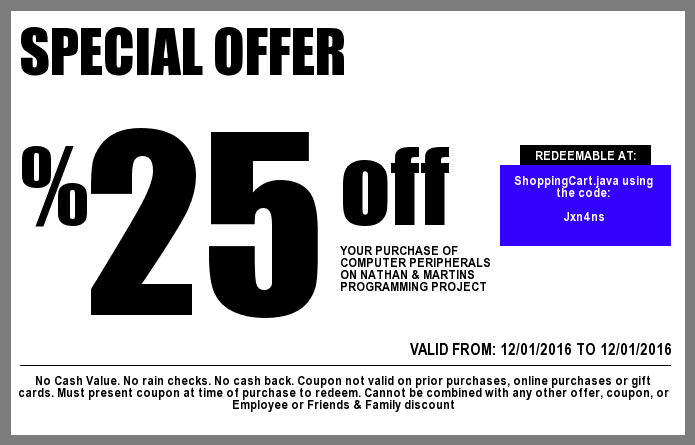
|  |
| --- |
| int index = -1;  Boolean wasFound = false;  for(int i = 0; i < array.length; i++)  if(search.equals(array[i]) && wasFound == false  index = i  wasFound = true  else if(search.equals(array[i]) && wasFound == true) Print out error that there are duplicates in the file Exit the program  return index; |

**ary.checkDuplicate**(String[] array)

|  |
| --- |
| for(int i = 0; i < array.length; i++)  for(int f = 0; f < array.length; f++)  if(array[i].equals(array[f]) && i != f)  Print out error that there are duplicates in the file  Exit the program |

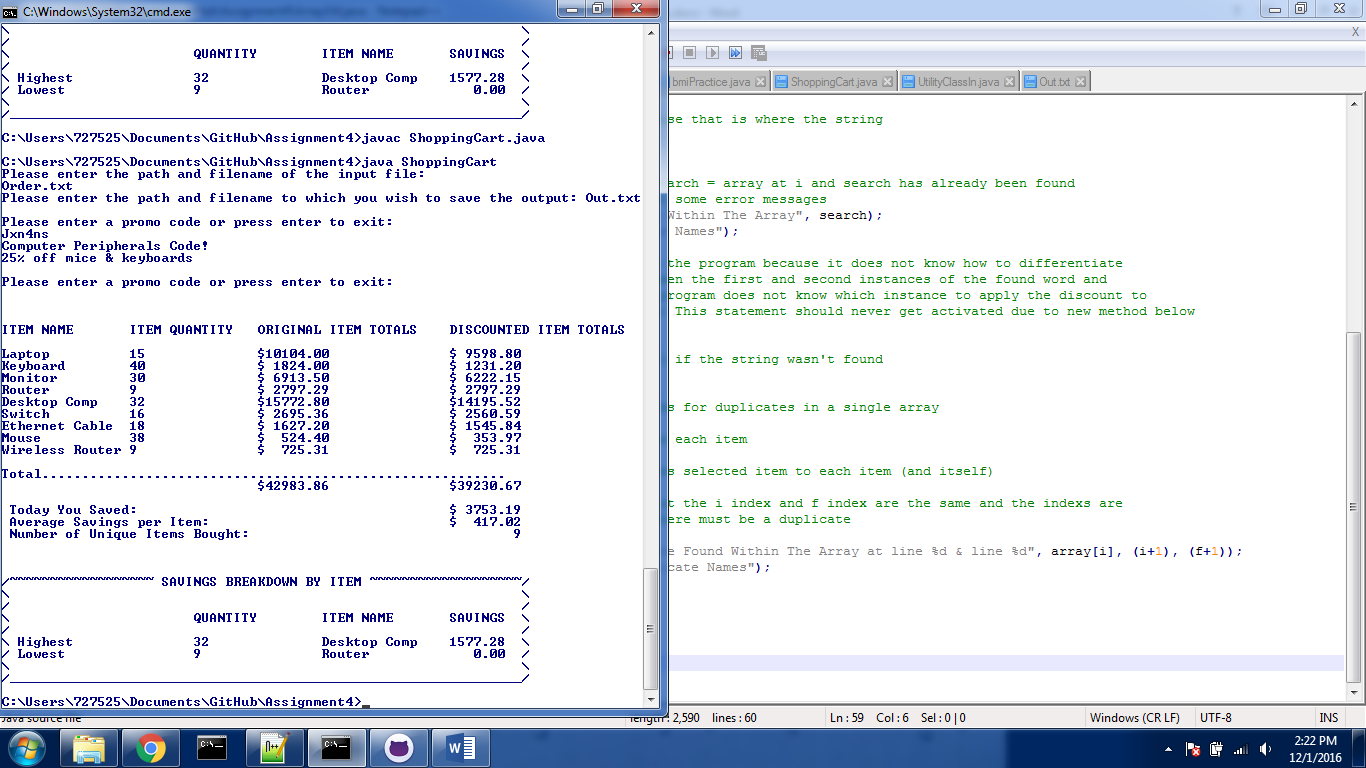
# Coupons





# Screenshots

Output with Test Data



Output with Zero Value in Price Column

