CS F222

Object Oriented Programming

Computer Science

Project

|  |  |
| --- | --- |
| Name | ID no. |
| Megha Manoj | 2020A7PS0016U |
| Ayush Mallick | 2020A7PS0257U |
| Aamina Taskeen | 2020A7PS0025U |
| Rohil Agarwal | 2020A7PS0259U |

**Birla Institute of Technology and Science**

**D.I.A.C, Dubai, UAE**

**A Report**

**on**

**Vending Machine**

**Prepared for**

**Dr Pranav**

**Instructor in charge**

**By**

|  |  |
| --- | --- |
| Name | ID no. |
| Megha Manoj | 2020A7PS0016U |
| Ayush Mallick | 2020A7PS0257U |
| Rohil Agarwal | 2020A7PS0259U |
| Aamina Taskeen | 2020A7PS0025U |

**Approved by**

**Dr. Pranav**

**Instructor in charge**

**December 2021**

**Abstract**

The main aim of this project is to showcase the application of the various concepts of java, learned in the Object-Oriented Programming course. The program has been designed with respect to the working of a vending machine where the customer inputs an item number and respective amount of cash to purchase a product. In this report, the code used for this particular concept has been explained with respect to the source code and its output has been showcased. The outputs indicate how the program reacts under various circumstances, say the wrong item number has been entered.

**Acknowledgement**

I would like to express my gratitude towards Dr. Pranav, the instructor- in charge of the course “Object Oriented Programming”, for this opportunity to deepen my conceptual understanding of object-oriented languages.

I would also like to express my deep sense of gratitude and indebtedness towards Dr. Sujala Shetty for her guidance and support throughout the project. Her teachings of the topics were fundamental in the completion of this project whilst being able to apply my creativity.

I am also thankful to Dr Vadivel for his advice and support in clearing queries on how the program could be improved.

**Table of contents**

|  |  |
| --- | --- |
| **Abstract** | **2** |
| **Acknowledgement** | **3** |
| 1. **Introduction**    1. **Vending Machine**    2. **Java** 2. **Code**     1. **Code explanation**    2. **Source code**    3. **Theory used** 3. **Output**     1. **Case 1**    2. **Case 2**    3. **Case 3**    4. **Case 4**    5. **Case 5**    6. **Case 6** 4. **Conclusion** | **5**  **5**  **5**  **6**  **6-7**  **7-25**  **26**  **27**  **27**  **28**  **29**  **30**  **31**  **32**  **33** |

**1.** **Introduction**

The main agenda of this report is to use the theoretical study of java that has been taught in class and apply it in a real-life application. We would be using the example of a vending machine to apply such theories and through the process understand how a code is formed and used.

**1.1.** **Vending machine**

A vending machine is an automated machine that provides items such as snacks, beverages and other essentials to a consumer after they make a payment in the form of cash or card . In a vending machine the system is coded in such a manner that when a customer inputs the payment and then selects the item, they would like to purchase the machine uses a roller and pushes the item selected into a collector tray from where the customer can get there item also if any excess money is left that is given out too.

**1.2.** **Java**

Java is an object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. The Java runtime provides dynamic capabilities such as runtime code modification that are typically not available in traditional compiled languages. It is user friendly and hence is widely used.

**2. Code**

**2.1. Code explanation**

A class Order has been created which takes the name of the item and price, which are available to the customer, through a constructor and sorts them using an overwritten compareTo method. The compareTo method is available for sorting by implementing Comparable. There are get and set methods for both the item as well as price as well as a ToString() method which returns the item and its corresponding price in a String format.

In method getItem(), it takes the item number and the payment from the user, and uses if-else statements to carry out the purchase. If the user has paid more than the required amount of money, a message is displayed which will mention the balance amount returned to the user from the payment.

The class main inherits the properties of class Order. It has a class constructor which takes the item and price and transfers the data to the parent class(Order) by using super(). The main method has a vector named List which consists of the vending machine options namely listing- the type of product, the order number for the product and its price in dirhams. The function Collections.sort() has been used to sort the vector List in increasing order of price using the compareTo() method as mentioned in class Order. The Scanner class object has been implemented for taking input of their order. A for-each loop has been applied which displays the menu items to the customer.

Using a while loop, the input is repeatedly taken from the user. A try catch block within the loop sees whether the item number entered by the user is correct by using an if-else statement. If the correct item number has been entered, then once the purchase is carried out, the program asks the customer if they wish to continue or not. If “yes” is entered, then the process repeats again, otherwise the program ends. The catch block checks for an InputMismatchException which, if found, terminates the loop.

**2.2. Source code**

import java.util.\*;

class Order implements Comparable<Order> {

public String item;

public double price;

public int compareTo(Order o1)

{

return item.compareTo(o1.item);

}

public Order(String item, double price)

{

super();

this.item = item;

this.price = price;

}

@Override

public String toString()

{

return "Item =" + item

+ " : Price =" + price + " AED";

}

public String getOrderNo()

{

return item;

}

public void setOrderNo(String item)

{

this.item = item;

}

public double getPrice()

{

return price;

}

public void setPrice(double price)

{

this.price = price;

}

public void getItem(String item,double price)

{

int u = Integer.valueOf(item);

if(((u == 1)||(u == 001)))

{

if(price == 6.00)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println("Water");

}else if(price>=6.00)

{

System.out.println("Heres ur change ["+(price - 6.00)+"]");

System.out.println("Water");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 2)||(u == 002)) == true)

{

if(price == 3.00)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Coke");

}else if(price>=3.00)

{

System.out.println("Heres ur change ["+(price - 3.00)+"]");

System.out.println( "Coke");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 3)||(u == 003)) == true )

{

if(price == 5.00)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "PPE - Stay Safe");

}else if(price>= 5.00)

{

System.out.println("Heres ur change ["+(price - 5.00)+"]");

System.out.println( "PPE - Stay Safe");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 4)||(u == 004)) == true)

{

if(price == 1.50)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "M&M's Chocolate");

}else if( price>=1.50)

{

System.out.println("Heres ur change ["+(price - 1.50)+"]");

System.out.println( "M&M's Chocolate");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 5)||(u == 005)) == true)

{

if(price == 4.75)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Lay's Onion and Cream");

}else if(price >= 4.75)

{

System.out.println("Heres ur change ["+(price - 4.75)+"]");

System.out.println( "Lay's Onion and Cream");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 6)||(u == 006)) == true)

{

if(price == 1.25)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Sanitiser");

}else if(price >= 1.25)

{

System.out.println("Heres ur change "+(price - 1.25));

System.out.println( "Sanitiser");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(((u == 7)||(u == 007)) == true)

{

if(price == 2.00)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Face Wipes");

}else if(price>=2.00)

{

System.out.println("Heres ur change ["+(price - 2.00)+"]");

System.out.println( "Face Wipes");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

if(u == 8)

{

if(price == 12.00)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Batteries");

}else if(price>= 12.00)

{

System.out.println("Heres ur change ["+(price - 12.00)+"]");

System.out.println( "Batteries");

}

else

{

System.out.println("Sorry not enough money, Heres ur money["+price+"]");

}

}

if(u == 9)

{

if(price == 6.95)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Cookies! Yay!");

}else if(price>=6.95)

{

System.out.println("Heres ur change ["+(price - 6.95)+"]");

System.out.println( "Cookies! Yay!");

}

else

{

System.out.println("Sorry not enough money, Heres ur money["+price+"]");

}

}

if(u == 10)

{

if(price == 3.75)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Brownies");

}else if(price>=3.75)

{

System.out.println("Heres ur change ["+(price - 3.75)+"]");

System.out.println( "Brownies");

}

else

{

System.out.println("Sorry not enough money, Heres ur money["+price+"]");

}

}

if(u == 11)

{

if(price == 4.25)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "Pringles");

}else if(price>= 4.25)

{

System.out.println("Heres ur change ["+(price - 4.25)+"]");

System.out.println( "Pringles");

}

else

{

System.out.println("Sorry not enough money, Heres ur money["+price+"]");

}

}

if(u == 12 )

{

if(price == 2.85)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "[Apple Juice]");

}else if(price>=2.85)

{

System.out.println("Heres ur change ["+(price - 2.85+"]"));

System.out.println( "[Apple Juice]");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

if(u == 13)

{

if(price == 3.42)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "[Mountain Dew]");

}else if(price>=3.42)

{

System.out.println("Heres ur change ["+(price - 3.43)+"]");

System.out.println( "[Mountain Dew]");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

if(u == 14 )

{

if(price == 12.50)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "[RedBull]");

}else if(price>= 12.50)

{

System.out.println("Heres ur change ["+(price - 12.50)+"]");

System.out.println( "[RedBull]");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

if(((u == 15) || (u == 16)))

{

if(price == 1.55)

{

System.out.println("You gave the exact amount, Thank You");

System.out.println( "[Muffin]");

}else if(price >= 1.55)

{

System.out.println("Heres ur change ["+(price - 1.55)+"]");

System.out.println( "[Muffin]");

}

else

{

System.out.println("Sorry not enough money, Heres ur money ["+price+"]");

}

}

else

{

System.out.println( "Make sure to collect ur change! Cheers!");

}

}

}

class main extends Order

{

main(String item, double price)

{

super(item,price);

}

public static void main(String[] args)

{

List<Order> list = new Vector<>();

list.add(new Order("Water 2L - ..................[1]", 6.00));

list.add(new Order("Coca-Cola - ...................[2]", 3.00));

list.add(new Order("PPE - .................[3]", 5.00));

list.add(new Order("M&M's Chocolate - ............[4]", 1.50));

list.add(new Order("Lay's Onion and Cream - .........[5]", 4.75));

list.add(new Order("Sanitiser - ................[6]", 1.25));

list.add(new Order("Face wipes - ................[7]", 2.00));

list.add(new Order("Removable Batteries (AAA) x3 - ..[8]", 12.00));

list.add(new Order("Cookies Can - ................[9]", 6.95));

list.add(new Order("Brownies x1 - ...............[10]", 3.75));

list.add(new Order("Pringles - ...............[11]", 4.25));

list.add(new Order("Apple Juice - ...............[12]", 2.85));

list.add(new Order("Mountain Dew - ..............[13]", 3.43));

list.add(new Order("RedBull - ..............[14]", 12.50));

list.add(new Order("Muffin - Pineapple - .........[15]", 1.55));

list.add(new Order("Muffin - Coconut - .......[16]", 1.55));

Collections.sort(list,Comparator.comparingDouble(Order::getPrice));

System.out.println("Welcome to the Vending Machine");

System.out.println("Here are the items we offer, What would you have? Enter the no. next to the item!(Integer input)");

Scanner sc = new Scanner(System.in);

for(Order o:list)

{

System.out.println(o.toString());

}

while(sc.hasNext())

{

try

{

int inputitem = sc.nextInt();

if(inputitem > 016)

{

System.out.println("Sorry the Product code doesnt exist!");

}

else

{

String inputItem = String.valueOf(inputitem);

Scanner sx = new Scanner(System.in);

System.out.println("Enter your money now!(Double input)");

double inputMoney = sx.nextDouble();

main obj = new main(inputItem,inputMoney);

obj.getItem(inputItem,inputMoney);

System.out.println("Would you like anything else? (Yes/No)");

Scanner st = new Scanner(System.in);

String ui = st.nextLine();

String uppercase = ui.toUpperCase();

if(uppercase.equals("NO"))

{

break;

}

}

}

catch(InputMismatchException e)

{

System.out.println("Sorry it seems like you didnt follow the Instructions, try again!");

break;

}

}

System.out.println("Peace :)!");

}

}

**2.3. Theory used**

the following are the theories used in the code

· comparator(comparable)

· class

· compareTo methods

· ToString method

· Sort method

· Super()

· Override method

· Void and return methods

· Constructor

· If else loop

· Nested loop

· Enhanced loop

· While loop

· Inheritance

· Vector

· Vector methods

· Set and get method (This keyword)

· Scanner class

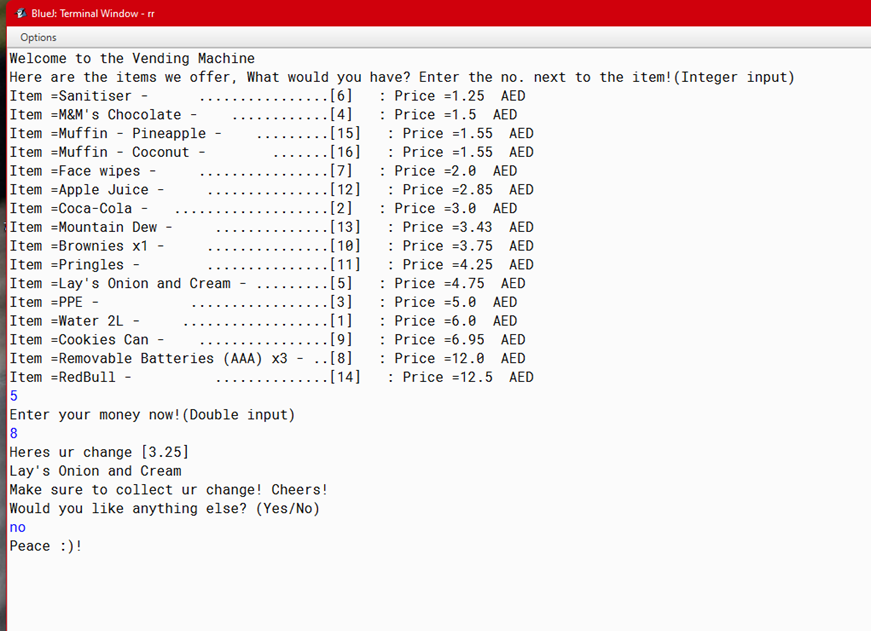
· Try-catch

**3. Output**

The following are output possibilities for the written code

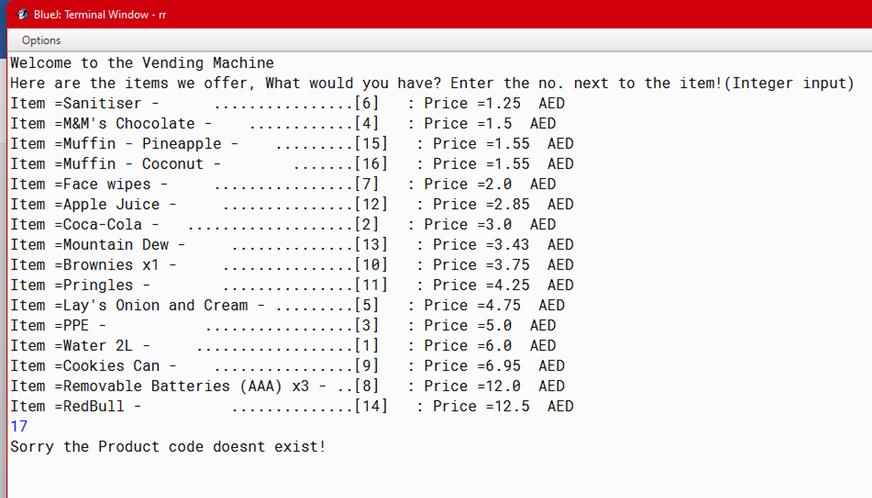
**3.1. Case 1**

If all the inputs are put correctly and only 1 item is to be brought by the user. The following is the output of such a case.

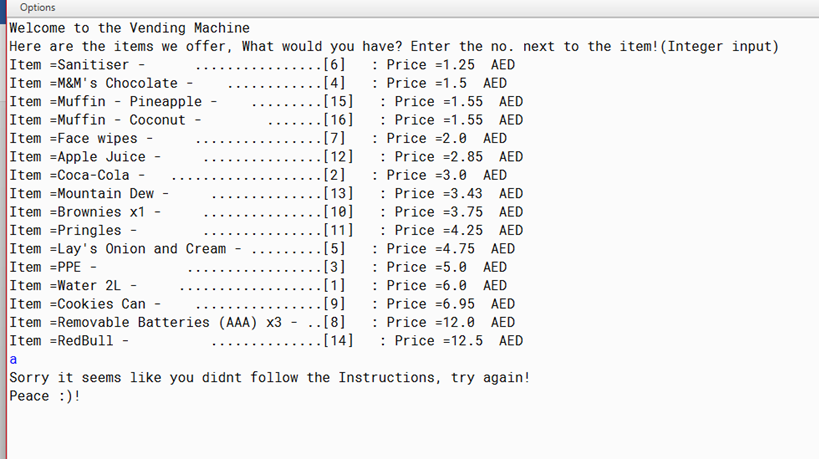


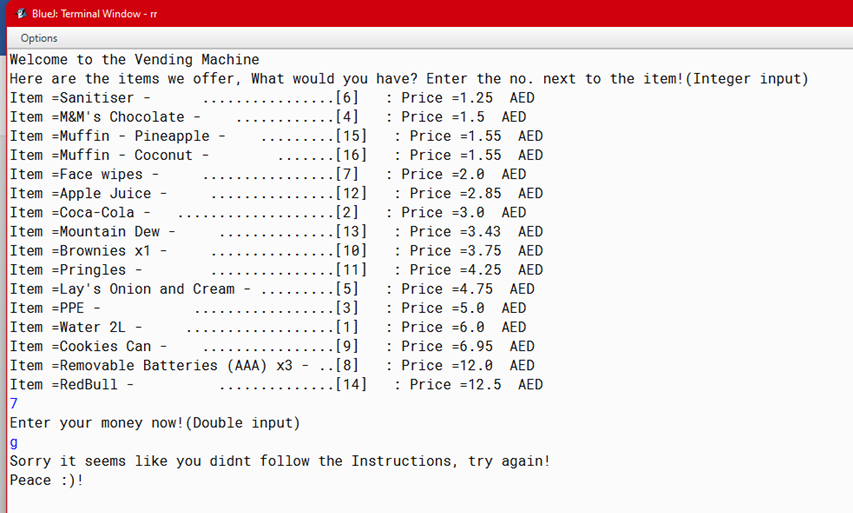
**3.2. Case 2**

If the input for the item code doesn’t exist in the system. The following is the output of such a case.

****

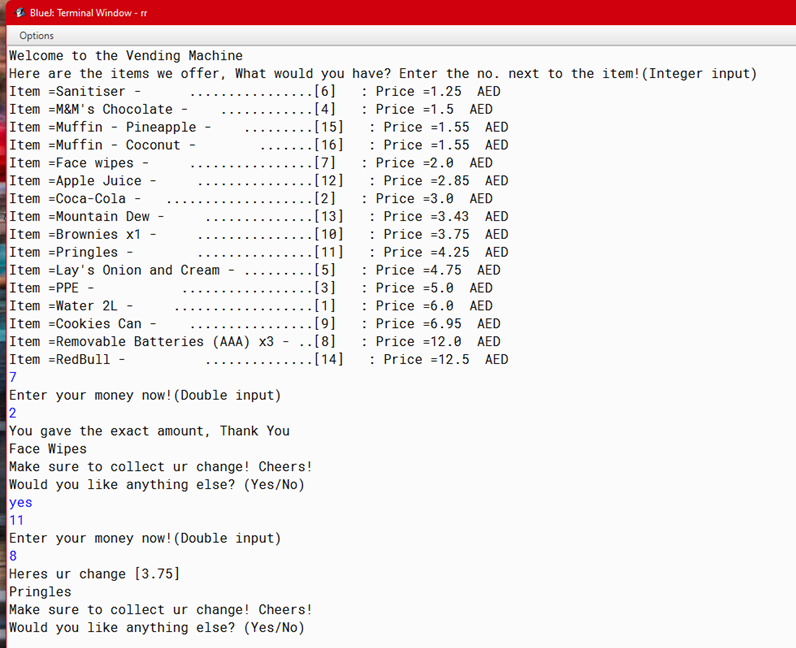
**3.3. Case 3**

If the input for any numerical value has been by mistake written as a character. The following is the output of such case****

****

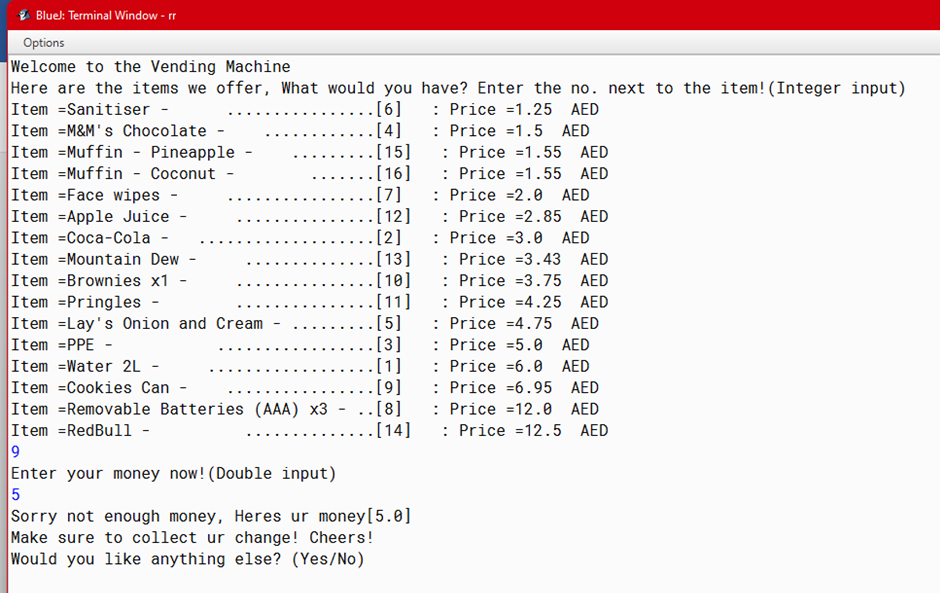
**3.4. Case 4**

If all the inputs are put correctly and 2 items are to be brought by the user. The following is the output of such a case.



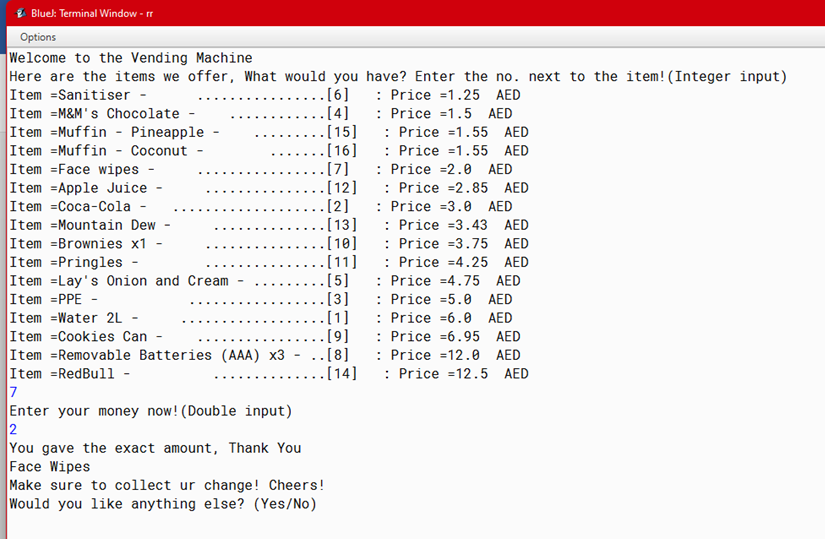
**3.5. Case 5**

If all the inputs are put correctly but the money put by the customer is below the price of the item. The following is the output of such a case.



**3.6. Case 6**

If all the inputs are put correctly and the money put by the customer is exactly the price of the item. The following is the output of such a case.



**4. Conclusion**

This report aims to explain to us how java concepts can be applied in real life scenarios. Here we have constructed a vending machine from scratch using the Java programming language. A vending machine is a machine that dispenses small articles such as food, drinks, or cigarettes when a coin or token is inserted.

We have defined the class for all of the products that are sold, the fields have been defined for each product to be dispensed, and the conditionals and loop fields have been used to enable the customers to navigate their way through the menu. The items of the machine have been sorted in an increasing order of item price. A balance is also returned to the customer if excess funds have been inserted. An exception is also thrown when a wrong input has been entered.

From the viewpoint of application, theories such as inheritance, nested loops, super, vectors, constructors, comparable, exceptions ,etc. have been used for the effective implementation of the code.

As for application, there are many places where Java is used in the real world, starting from a commercial e-commerce website to android apps, from scientific application to financial applications like electronic trading systems, from games like Minecraft to desktop applications like Eclipse, Netbeans, etc. This shows there is no limit to java as it can serve our every need.