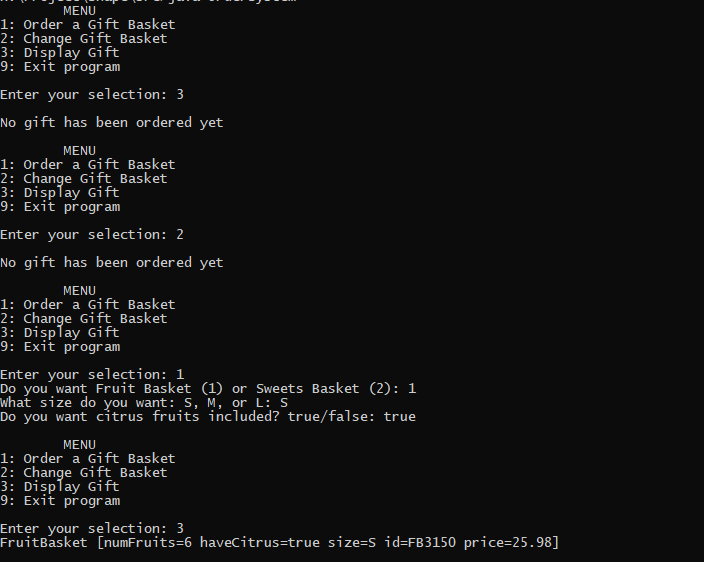
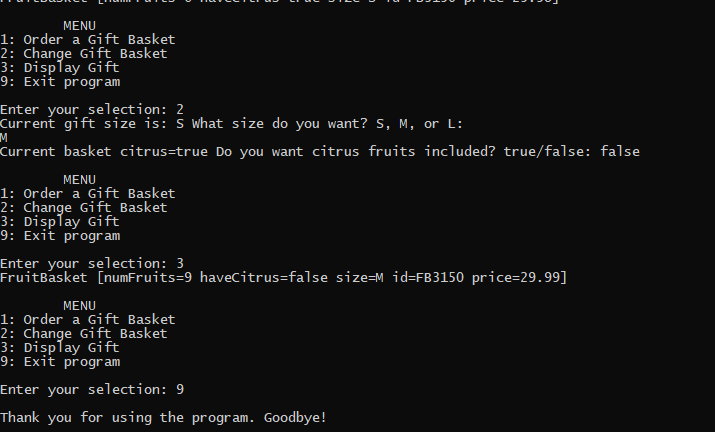
**Output Screen Shot:**





**Program Code [Java]**

**Gift.java**

/\*\*

\* The type Gift.

\*/

public class Gift {

// All required attributes

/\*\*

\* The Id.

\*/

protected final String id; // Its value cannot be changed so make it final

/\*\*

\* The Size.

\*/

protected char size;

/\*\*

\* The Price.

\*/

protected double price;

// Constructor except price

/\*\*

\* Instantiates a new Gift.

\*

\* @param id the id

\* @param size the size

\*/

public Gift(String id, char size) {

this.id = id;

this.size = size;

// Calculating price depending on size calculate price

calculatePrice();

}

// Setters & Getters

/\*\*

\* Gets id.

\*

\* @return the id

\*/

public String getId() {

return id;

}

/\*\*

\* Gets size.

\*

\* @return the size

\*/

public char getSize() {

return size;

}

/\*\*

\* Sets size.

\*

\* @param size the size

\*/

public void setSize(char size) {

this.size = size;

}

// Getter for price

/\*\*

\* Gets price.

\*

\* @return the price

\*/

public double getPrice() {

return price;

}

// A method which return class values as a String

@Override

public String toString() {

return "size=" + size + " id=" + id + " price=" + String.format("%.2f", price) + "]";

}

// calculatePrice method

/\*\*

\* Calculate price.

\*/

public void calculatePrice() {

if (size == 'S') {

price = 19.99;

}

else if (size == 'M') {

price = 29.99;

}

else if (size == 'L') {

price = 39.99;

}

}

}

**FruitBasket.java**

/\*\*

\* The type Fruit basket.

\*/

public class FruitBasket extends Gift{

private int numOfFruits;

private boolean hasCitrus;

// Constructor, Setters & Getters

/\*\*

\* Instantiates a new Fruit basket.

\*

\* @param id the id

\* @param size the size

\* @param isCitrus the is citrus

\*/

public FruitBasket(String id, char size, boolean isCitrus) {

super(id, size);

this.hasCitrus = isCitrus;

// Depending on size, assigning number of fruits

// Add additional fee of 5.99 if Fruit basket gift has citrus

calculateNumOfFruits();

calculatePrice();

}

/\*\*

\* Is citrus boolean.

\*

\* @return the boolean

\*/

public boolean isCitrus() {

return hasCitrus;

}

/\*\*

\* Sets citrus.

\*

\* @param citrus the citrus

\*/

public void setCitrus(boolean citrus) {

hasCitrus = citrus;

}

// toString method

@Override

public String toString() {

return "FruitBasket [numFruits=" + numOfFruits + " haveCitrus=" + hasCitrus + " " + super.toString();

}

@Override

public void calculatePrice() {

super.calculatePrice();

if (hasCitrus)

this.price += 5.99;

}

// calculateNumOfFruits method

/\*\*

\* Calculate num of fruits.

\*/

public void calculateNumOfFruits() {

if (size == 'S')

numOfFruits = 6;

else if (size == 'M')

numOfFruits = 9;

else if (size == 'L')

numOfFruits = 15;

}

}

**SweetsBasket.java**

/\*\*

\* The type Sweets basket.

\*/

public class SweetsBasket extends Gift{

// Required attributes

private boolean hasNuts;

// Constructor

/\*\*

\* Instantiates a new Sweets basket.

\*

\* @param id the id

\* @param size the size

\* @param hasNuts the has nuts

\*/

public SweetsBasket(String id, char size, boolean hasNuts) {

super(id, size);

this.hasNuts = hasNuts;

}

// Setters & Getters

/\*\*

\* Is has nuts boolean.

\*

\* @return the boolean

\*/

public boolean isHasNuts() {

return hasNuts;

}

/\*\*

\* Sets has nuts.

\*

\* @param hasNuts the has nuts

\*/

public void setHasNuts(boolean hasNuts) {

this.hasNuts = hasNuts;

}

@Override

public String toString() {

return "SweetBasket [hasNuts=" + hasNuts + " " + super.toString();

}

}

**OrderSystem.java**

import java.util.Scanner;

// Order System class

/\*\*

\* The type Order system.

\*/

public class OrderSystem {

/\*\*

\* The entry point of application.

\*

\* @param args the input arguments

\*/

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

// Creating an instance of gift

Gift gift = null;

int basketChoice = 0;

// main method which would have all the functionality mentioned

while (true) {

// Showing menu

System.out.println("\tMENU");

System.out.println("1: Order a Gift Basket");

System.out.println("2: Change Gift Basket");

System.out.println("3: Display Gift");

System.out.println("9: Exit program");

System.out.println();

System.out.print("Enter your selection: ");

int opt = scan.nextInt();

if (opt == 1) {

// Asking for choice

char size;

boolean hasNC;

System.out.print("Do you want Fruit Basket (1) or Sweets Basket (2): ");

basketChoice = scan.nextInt();

System.out.print("What size do you want: S, M, or L: ");

size = scan.next().charAt(0);

// Depending on choice of basket, ask for required information

if (basketChoice == 1) {

System.out.print("Do you want citrus fruits included? true/false: ");

hasNC = scan.nextBoolean();

gift = new FruitBasket("FB3150", size, hasNC);

}

else if (basketChoice == 2) {

System.out.println("Do you want nuts included? true/false: ");

hasNC = scan.nextBoolean();

gift = new SweetsBasket("FB3150", size, hasNC);

}

}

else if (opt == 2) {

if (gift == null) {

System.out.println("\nNo gift has been ordered yet");

}

else {

// Else asking for changes in the order

System.out.println("Current gift size is: " + gift.getSize() + " What size do you want? S, M, or L:");

char size = scan.next().charAt(0);

boolean hasNC;

gift.setSize(size);

if (basketChoice == 1) {

System.out.print("Current basket citrus=" + ((FruitBasket)gift).isCitrus() + " Do you want citrus fruits included? true/false: ");

hasNC = scan.nextBoolean();

((FruitBasket) gift).setCitrus(hasNC);

((FruitBasket) gift).calculateNumOfFruits();

}

else if (basketChoice == 2) {

System.out.print("Current basket nuts=" + ((SweetsBasket)gift).isHasNuts() + " Do you want nuts included? true/false: ");

hasNC = scan.nextBoolean();

((SweetsBasket) gift).setHasNuts(hasNC);

}

gift.calculatePrice();

}

}

else if (opt == 3) {

if (gift == null) {

System.out.println("\nNo gift has been ordered yet");

}

else {

// Displaying Gift

System.out.println(gift);

}

}

else if (opt == 9) {

System.out.println();

break;

}

System.out.println();

}

System.out.println("Thank you for using the program. Goodbye!");

}

}