American International University-Bangladesh

TEAM

Student Name	Student Id
MOHAIMENUR RAHMAN	19-40338-1
JANNATUL FERDOUS NISHI	19-40374-1

Project Title: Food Court Management Application.

Classes

Appitizers

```
package project;

public class Appitizers extends FoodItem {

private String size;

        public void setSize(String size)
        {this.size=size;}
        public String getSize()
        {return size;}
        public void showInfo()
        {
                System.out.println("Food ID: "+getFid());
                System.out.println("Food Name: "+getName());
                System.out.println("Available Quantity: "+getAvailableQuantity());
                System.out.println("Food price :"+getPrice());
          System.out.println("Size of the food:"+getSize());
                System.out.println();
```

```
            }

}
```

Employee

```
package project;
import java.lang.*;

public class Employee {

            private String name;
            private String empId;
            private double salary;

            public void setName(String name)
            {this.name = name;        }
            public void setEmpId(String empId)
            {this.empId = empId;}
            public void setSalary(double salary)
            {this.salary = salary;}

            public String getEmpId()
            {return empId;}
            public String getName()
            {return name;}
            public double getSalary()
            {return salary;}

}
```

FoodCourt

```
package project;
import java.lang.*;
public class FoodCourt implements RestaurantOperations,EmployeeOperations {

            private Restaurant restaurants[] = new Restaurant[100];
            private Employee employees[] = new Employee[200];

 public void insertRestaurant(Restaurant r){
    int flag=0;
    for(int i=0;i<restaurants.length;i++)
    {

      if(restaurants[i]==null)
      {
        restaurants[i]=r;
        flag=1;
        break;
      }
    }
```

```
  if(flag==1){System.out.println("Restaurant inserted");}
  else{System.out.println("Restaurant Can not be inserted");}
}

      public void removeRestaurant(Restaurant r)
{
  int flag = 0;
            for(int i=0; i<restaurants.length; i++)
            {
                  if(restaurants[i] == r)
                  {
                        restaurants[i] = null;
                        flag = 1;
                        break;
                  }
            }
            if(flag == 1){System.out.println("Restaurant Removed");}
            else{System.out.println("Resturant can Not be Removed");}
}
 public Restaurant getRestaurant(String rid){
  Restaurant r = null;
            for(int i=0; i<restaurants.length; i++)
            {
                  if(restaurants[i] != null)
                  {
                        if(restaurants[i].getRid().equals(rid)) {
                              r = restaurants[i];
                              break;
                        }
                  }
            }
            return r;
}

public void showAllRestaurants(){
            for(Restaurant r : restaurants){
                  if(r!= null)
                  {
                        System.out.println("**********************************");
                        System.out.println("Resturant Name: "+ r.getName());
                        System.out.println("Resturant ID: "+ r.getRid());
                        System.out.println("----------------------------------");
                        r.showAllFoodItems();
                        System.out.println("----------------------------------");
                        System.out.println();
                  }
            }
      }
      public void insertEmployee(Employee e)
      {
            int flag = 0;

            for(int i=0; i<employees.length; i++)
            {
                  if(employees[i] == null)
                  {
```

```
                employees[i] = e;
                flag = 1;
                break;
            }
        }
        if(flag == 1){System.out.println("Employee Inserted");}
        else{System.out.println("Employee can Not be Inserted");}
    }

    public void removeEmployee(Employee e)
    {
        int flag = 0;
        for(int i=0; i<employees.length; i++)
        {
            if(employees[i] == e)
            {
                employees[i] = null;
                flag = 1;
                break;
            }
        }
        if(flag == 1){System.out.println("Employee Removed");}
        else{System.out.println("Employee can Not be Removed");}
    }
    public Employee getEmployee(String empId)
    {
        Employee e = null;

        for(int i=0; i<employees.length; i++)
        {
            if(employees[i] != null)
            {
                if(employees[i].getEmpId().equals(empId))
                {
                    e = employees[i];
                    break;
                }
            }
        }
        return e;
    }
    public void showAllEmployees()
    {
        System.out.println("/////////////////////////////////");
        for(Employee e : employees)
        {
            if(e!= null)
            {
                System.out.println("Employee Name: "+ e.getName());
                System.out.println("Employee Id: "+ e.getEmpId());
                System.out.println("Salary: "+ e.getSalary());
                System.out.println();
            }
        }
        System.out.println("/////////////////////////////////");
    }
```

```
}
```

FoodItem

```
package project;
import java.lang.*;

public abstract class FoodItem implements IQuantity{

        protected String fid;
        protected String name;
        protected int availableQuantity;
        protected double price;


        public void setFid(String fid){this.fid=fid;}
        public void setName(String name){this.name=name;}
        public void setAvailableQuantity(int
availableQuantity){this.availableQuantity=availableQuantity;}
        public void setPrice(double price){this.price=price;}


        public String getFid(){return fid;}
        public String getName(){return name;}
        public int getAvailableQuantity(){return availableQuantity;}
        public double getPrice(){return price;}

        public abstract void showInfo();

        public void addQuantity(int amount)
        {
                if(amount>0)
                {
                        System.out.println("Previous Quantity: "+ availableQuantity);
                        System.out.println("Add Quantity: "+ amount);
                        availableQuantity += amount;
                        System.out.println("Current Quantity: "+ availableQuantity);
                }
                else
                {
                        System.out.println("Can Not Add Quantity");
                }
        }
        public void sellQuantity(int amount)
        {
                if(amount>0 && amount<=availableQuantity)
                {
                        System.out.println("Previous Quantity:   "+ availableQuantity);
                        System.out.println("Sell Quantity:       "+ amount);
                        availableQuantity -= amount;
                        System.out.println("Current Quantity:    "+ availableQuantity);
                }
        else
        {
            System.out.println("Can Not sell ");
        }
    }



}
```

MainDish

```
package project;
import java.lang.*;

public class MainDish extends FoodItem {

private String category;

    public void setCategory(String category)
    {
        this.category=category;
    }
    public String getCategory()
    {
        return category;
    }
    public void showInfo()
    {
        System.out.println("Food ID: "+getFid());
        System.out.println("Food Name: "+getName());
        System.out.println("Available Quantity: "+getAvailableQuantity());
        System.out.println("Food price :"+getPrice());
     System.out.println("Food category:"+getCategory());
        System.out.println();
    }



}
```

Restaurant

```
package project;

import java.lang.*;


import fp.FoodItem;
```

```
public class Restaurant implements FoodItemOperations {


    private String rid;
    private String name;
    private FoodItem foodItems[] = new FoodItem[10];

    public void setRid(String rid){this.rid = rid;}
    public void setName(String name){this.name = name;}



    public String getRid(){return rid;}
    public String getName(){return name;}



     public void insertFoodItem(FoodItem f){
 int flag = 0;
        for(int i=0; i<foodItems.length; i++){

            if(foodItems[i] == null){
                foodItems[i] = f;
                flag = 1;
                break;
            }
        }
        if(flag == 1)
        {
            System.out.println("Food  Inserted");
        }
        else
        {
            System.out.println("Can Not Insert");
        }
}

public void removeFoodItem(FoodItem f){
 int flag = 0;
            for(int i=0; i<foodItems.length; i++){
                   if(foodItems[i] == f){
                         foodItems[i] = null;
                         flag = 1;
                         break;
  }
 }
 if(flag == 1){
  System.out.println("Food  has been Removed");
 }
            else{
  System.out.println("Can Not Remove");
 }
}

public FoodItem getFoodItem(String fid){
            FoodItem f = null;
            for(int i=0; i<foodItems.length; i++){
                   if(foodItems[i] != null){
                         if(foodItems[i].getFid().equals(fid) ){
                                f = foodItems[i];
                                break;
                         }
                   }
            }
            return f;
      }
 public void showAllFoodItems(){

  for(FoodItem f : foodItems){

                    if(f != null){

                           f.showInfo();

                    }

             }

 }



}
```

interfaces

EmployeeOperations

```
package project;
import java.lang.*;

public interface EmployeeOperations {

       void insertEmployee(Employee e);
       void removeEmployee(Employee e);
       Employee getEmployee(String empId);
       void showAllEmployees();

}
```

FoodItemOperations

```
package project;

import java.lang.*;

import fp.FoodItem;
```

```
public interface FoodItemOperations {


	void insertFoodItem(FoodItem f);

	void removeFoodItem(FoodItem f);

	FoodItem getFoodItem(String fid);

	void showAllFoodItems();


}
```

IQuantity

```
package project;
import java.lang.*;

public interface IQuantity {

	void addQuantity(int amount);
	void sellQuantity(int amount);
}
```

RestaurantOperations

```
package project;
import java.lang.*;

public interface RestaurantOperations {
	void insertRestaurant(Restaurant r);
	void removeRestaurant(Restaurant r);
	Restaurant getRestaurant(String rid);
	void showAllRestaurants();

}
```

Fileio

FileReadWriteDemo

```
package project;
```

```
import java.lang.*;

import java.io.*;


public class FileReadWriteDemo {


    private File file;                          //to create a File

    private FileWriter writer;          //to write in a file

    private FileReader reader;                  //to read from a file

    private BufferedReader bfr;                 //to read file content as a String




    public void writeInFile(String s)

    {

        /*

            creating a file and writing in it genarates compile time exceptions (Checked
Exceptions).

            So, we need to write the whole thing in try-catch.

        */

        try

        {

            file = new File("History.txt");                     //Declaring a file named named
History.txt for creating.

            file.createNewFile();                                       //If the file does not
exists, creates and opens the file. else, just opens the file

            writer = new FileWriter(file, true);        //creating the writer object to write in the
file.

            writer.write(s+"\r"+"\n");                          //writing a string s in the file. the
"\r" and "\n" has been concat to go to a newline.

            writer.flush();                                                     //After writing,
we need to flush to indicate that we have completed writing.

            writer.close();                                                     //After flushing,
we need to close the file to save our writing.

        }

        catch(IOException ioe)
        {
            ioe.printStackTrace();
        }
    }

    public void readFromFile()
    {
        /*
            reading from a file genarates compile time exceptions (Checked Exceptions).
            So, we need to write the whole thing in try-catch.
        */

        try
        {
            reader = new FileReader(file);              //creating the reader object to read from a file.
            bfr = new BufferedReader(reader);           //creating the BufferedReader object using the reader object to read the file content.
            String text="", temp;                            //declaring two string variables to read the file content and storing them.

            while((temp=bfr.readLine())!=null)          //reading one line from the file, storing it in the variable temp and checking whether it is null or not. It will be null at the end of reading from the file.
            {
                text=text+temp+"\n"+"\r";                    //storing the temp string in text by concating it with text and "n" and "\r" is used to go to a newline.
            }

            System.out.print(text);                     //printing the whole string in console.
            reader.close();                                      //closing the file.
        }
        catch(IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
```

Start

```
package project;
import java.lang.*;
import java.util.*;
import fp.Appitizers;
import fp.Employee;
import fp.FileReadWriteDemo;
import fp.FoodCourt;
import fp.FoodItem;
import fp.MainDish;
import fp.Restaurant;

public class Start {

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        FoodCourt fc = new FoodCourt();
        FileReadWriteDemo frwd = new FileReadWriteDemo();

        boolean choice = true;

        try{
```

```
while(choice)
   {
        System.out.println("Choose from the Following Options: ");
        System.out.println("-------------------------------------");
        System.out.println("1. Employee Management");
        System.out.println("2. Restaurant Management");
        System.out.println("3. Restaurant Food Item Management");
        System.out.println("4. Food Item Quantity Add-Sell");
        System.out.println("5. Exit");
        System.out.println("-------------------------------------\n");

        System.out.print("You have choosed: ");

        int option = sc.nextInt();

        switch(option)
        {
                case 1:
                   try{

                        System.out.println("********************");
                        System.out.println("Employee Management");
                        System.out.println("What do you want to do?\n");
                        System.out.println("-------------------------------------");
                        System.out.println("1. Insert New Employee");
                        System.out.println("2. Remove an Existing Employee");
                        System.out.println("3. Show All Employees");
                        System.out.println("4. Search An Employee");
                        System.out.println("5. Go Back");
                        System.out.println("-------------------------------------\n");
                        System.out.print("Enter your option: ");
                int input1 = sc.nextInt();

                switch(input1)
                {
                    case 1:
                      try{


System.out.println("#####################");

                        System.out.println("Insert New Employee");

                        System.out.print("Enter Employee Name: ");
                        String name1 = sc.next();

                        System.out.print("Enter Employee ID: ");
                        String empId1 = sc.next();

                        System.out.print("Enter Employee Salary: ");
                        double salary = sc.nextDouble();

                        Employee e1 = new Employee();
                        e1.setEmpId(empId1);
                        e1.setName(name1);
                        e1.setSalary(salary);

                        fc.insertEmployee(e1);

                        System.out.println("#####################");
                        break;
                         }
```

```
                                      catch (InputMismatchException ime7)
                       {
                                                     ime7.printStackTrace();

       System.out.println("InputMismatchException occured");
                                 }




                                        case 2:
                                          try{

                                          System.out.println("#####################");
                                              System.out.println("Remove Existing
Employee");


                                              System.out.print("Enter Employee ID: ");
                                              String empId2 = sc.next();

                                              Employee e2 = fc.getEmployee(empId2);
                                              fc.removeEmployee(e2);




                                              System.out.println("#####################");
                                              break;
                                              }

                                      catch (InputMismatchException ime8)
                       {
                                                     ime8.printStackTrace();

       System.out.println("InputMismatchException occured");
```

```
                                       }

                                                  case 3:
                                                     try{
                                                            System.out.println("#####################");
                                                            System.out.println("Show All Employees");

                                                            fc.showAllEmployees();
                                                            System.out.println("#####################");
                                                            break;
                                                             }
                                               catch (InputMismatchException ime9)
                         {
                                                                      ime9.printStackTrace();

System.out.println("InputMismatchException occured");
                                       }


                                                  case 4:
                                                     try{
                                                      System.out.println("#####################");
                                                            System.out.println("Search An Employee");

                                                            System.out.print("Enter Employee ID: ");
                                                            String empId3 = sc.next();

                                                            Employee e3 = fc.getEmployee(empId3);

                                                            if(e3 !=null)
                                                            {
```

```
                                                            System.out.println("Employee Name:
"+e3.getName());
                                                            System.out.println("Employee ID:
"+e3.getEmpId());
                                                            System.out.println("Employee Salary:
"+e3.getSalary());
                                                    }
                                                    else
                                                    {
                                                            System.out.println("Employee Does Not
Exists");
                                                    }
                                                    System.out.println("#####################");
                                                    break;
                }
                                            catch (InputMismatchException ime10)
                        {
                                                            ime10.printStackTrace();

      System.out.println("InputMismatchException occured");
                                    }



                                              case 5:



                                                    System.out.println("#####################");
                                                    System.out.println("Go Back");
                                                    System.out.println("#####################");
                                                    break;

                                              default:
                                                    System.out.println("#####################");
```

```
                                                  System.out.println("Invaild Choice");
                                                  System.out.println("####################");
                                                  break;
                                 }
                                 System.out.println("*******************");
                                 break;
                                 }
                                  catch (InputMismatchException ime2)
                {
                                                          ime2.printStackTrace();

System.out.println("InputMismatchException occured");
                           }



                         case 2:
                            try{


                                 System.out.println("*******************");
                                 System.out.println("Restaurant Management");
                                 System.out.println("What do you want to do?\n");
                                 System.out.println("-------------------------------------");
                                 System.out.println("1. Insert New Restaurant");
                                 System.out.println("2. Remove an Existing Restaurant");
                                 System.out.println("3. Show All Restaurants");
                                 System.out.println("4. Search a Restaurant");
                                 System.out.println("5. Go Back");
                                 System.out.println("-------------------------------------\n");
                                 System.out.print("Enter your option: ");
                                 int input2 = sc.nextInt();

                                 switch(input2)
                {
                    case 1:
                        try{
                        System.out.println("#####################");
                        System.out.println("Insert New Restaurant");

                        System.out.print("Enter Restaurant Rid: ");
                        String rid1 = sc.next();

                        System.out.print("Enter Restaurant Name: ");
                        String name1 = sc.next();

                        Restaurant r1 = new Restaurant();
                        r1.setRid(rid1);
                        r1.setName(name1);

                        fc.insertRestaurant(r1);

                        System.out.println("#####################");
                        break;
                         }
                  catch (InputMismatchException ime11)
          {
                            ime11.printStackTrace();

System.out.println("InputMismatchException occured");
                }



                    case 2:
                        try {
```

```
                                        System.out.println("#####################");
                                        System.out.println("Remove An Existing
Restaurant");

                                        System.out.print("Enter Restaurant RID: ");
                                        String rid2 = sc.next();

                                        Restaurant r2 = fc.getRestaurant(rid2);
                                        fc.removeRestaurant(r2);



                                        System.out.println("#####################");
                                        break;
                                         }
                                  catch (InputMismatchException ime12)
                        {
                                            ime12.printStackTrace();

     System.out.println("InputMismatchException occured");
                              }

                                   case 3:
                                        try{
                                        System.out.println("#####################");
                                        System.out.println("Show All Restaurants");

                                        fc.showAllRestaurants();
                                        System.out.println("#####################");
                                        break;
                                         }
                                 catch (InputMismatchException ime13)
                        {
                                            ime13.printStackTrace();

    System.out.println("InputMismatchException occured");
                              }

                                   case 4:
                                        try{
                                        System.out.println("########################");
                                        System.out.println("Search a restaurant :");

                                        System.out.print("Enter Restaurant rid: ");
                                        String rid3 = sc.next();

                                        Restaurant r3 = fc.getRestaurant(rid3);

                                        if(r3 !=null)
                                        {
                                            System.out.println("Restaurant RID: 
"+r3.getRid());
                                            System.out.println("Restaurant Name: 
"+r3.getName());
                                            r3.showAllFoodItems();
                                        }
                                        else
                                        {
                                            System.out.println("Restaurant Does 
Not Exist");
                                        }
                                        System.out.println("########################");
                                        break;
                                         }
                                 catch (InputMismatchException ime14)
```

```
                   {
                                                     ime14.printStackTrace();

System.out.println("InputMismatchException occured");
                              }

                                      case 5:

                                              System.out.println("#####################");
                                              System.out.println("Go Back");
                                              System.out.println("#####################");
                                              break;

                                      default:
                                              System.out.println("#####################");
                                              System.out.println("Invaild Choice");
                                              System.out.println("#####################");
                                              break;
                               }
                               System.out.println("********************");
                               break;
                                }
                                 catch (InputMismatchException ime3)
               {
                                                     ime3.printStackTrace();

System.out.println("InputMismatchException occured");
                          }

                       case 3:
                          try{
```

```
                        System.out.println("********************");
                        System.out.println("Restaurant Food Item Management");
                        System.out.println("What do you want to do?\n");
                        System.out.println("----------------------------------------");
                        System.out.println("1. Insert New Food Item For Restaurant");
                        System.out.println("2. Remove An Existing Food Item Of A
Restaurant");
                        System.out.println("3. Show All Food Items Of A Restaurant");
                        System.out.println("4. Search A Food Item of a restaurant ");
                        System.out.println("5. Go Back");
                        System.out.println("----------------------------------------\n");
                        System.out.print("Enter your option: ");
                        int input3 = sc.nextInt();

                        switch(input3)
                        {
                            case 1:
                              try{
                                System.out.println("####################");
                                System.out.println("Insert New Food Item of a
Restaurant\n");


                                System.out.println("Which Type of Food Item Do
you Want to Add?");
                                System.out.println("1. Main Dish");
                                System.out.println("2. Appitizers");
                                System.out.println("3. Go Back");

                                System.out.print("Your Type: ");
                                int type = sc.nextInt();

                                FoodItem f = null;
```

```
                if(type == 1)
                {
                        System.out.print("Enter  FID Of Food: ");
                        String fid1 = sc.next();
                        System.out.print("Enter  Name Of Food:
");
                        String name1= sc.next();
                        System.out.print("Enter Available
Quantity: ");
                        int aq1 = sc.nextInt();
                        System.out.print("Enter Price Of The
Food: ");
                        double price1 = sc.nextDouble();
                        System.out.print("Enter  Category Of
The Food: ");
                        String category1= sc.next();


                        MainDish md = new MainDish();

                        md.setFid(fid1);
                        md.setName(name1);
                        md.setAvailableQuantity(aq1);
                        md.setPrice(price1);
                        md.setCategory(category1);

                        f = md;
                }
                else if(type == 2)
                {
                        System.out.print("Enter  FID Of Food: ");
                        String fid2 = sc.next();
                        System.out.print("Enter  Name Of Food:
");
```

```
                                        String name2= sc.next();
                                        System.out.print("Enter Available
Quantity: ");

                                        int aq2= sc.nextInt();
                                        System.out.print("Enter Price Of The
Food: ");

                                        double price2 = sc.nextDouble();
                                        System.out.print("Enter  Size Of The
Food: ");

                                        String size2= sc.next();



                                        Appitizers ap = new Appitizers();


                                        ap.setFid(fid2);
                                        ap.setName(name2);
                                        ap.setAvailableQuantity(aq2);
                                        ap.setPrice(price2);
                                        ap.setSize(size2);


                                        f = ap;
                                    }
                                    else if(type == 3)
                                    {
                                        System.out.println("Going Back....");
                                    }
                                    else
                                    {
                                        System.out.println("Invalid Type");
                                    }



                                    if(f != null)
```

```
                                        {
                                            try{
                                            System.out.print("Enter Restaurant RID: 
");
                                            String rid1 = sc.next();


                                            fc.getRestaurant(rid1).insertFoodItem(f);



                                            }
                                            catch(NullPointerException npe)
                                            {npe.printStackTrace();
                                            System.out.println("RID not found");
                                            }
                                        }

                                        System.out.println("#####################");
                                        break;
                                         }
                                  catch (InputMismatchException ime15)
                        {
                                            ime15.printStackTrace();

    System.out.println("InputMismatchException occured");
                               }



                                    case 2:


                                    try{
                                        System.out.println("#####################");
                                        System.out.println("Remove an Existing Food 
Item of a Restaurant");
```

```
                            System.out.print("Enter Restaurant RID: ");
                            String rid2 = sc.next();
                            System.out.print("Enter Food Item FID: ");
                            String fid2 = sc.next();




                            Restaurant rr = fc.getRestaurant(rid2);
                                FoodItem fi= rr.getFoodItem(fid2);

                                if (rr == null) {
                                    System.out.println("Food Item
FID Not Found!");
                                    break;
                                }
                                rr.removeFoodItem(fi);
                            }
                            catch (NullPointerException npe){
                                npe.printStackTrace();
                                System.out.println("NullPointerException
occured");
                            }

        System.out.println("#####################");
        break;
                        case 3:
                          try{
                            System.out.println("#####################");
                            System.out.println("Show All Food Items of a
Restaurant");
                            System.out.print("Enter Restaurant RID: ");
                            String rid3 = sc.next();
                            fc.getRestaurant(rid3).showAllFoodItems();
                             }

                      catch (InputMismatchException ime16)
              {
                                ime16.printStackTrace();

    System.out.println("InputMismatchException occured");
                    }
                            System.out.println("#####################");
                            break;
                        case 4:
                         try{
                            System.out.println("#####################");
                            System.out.println("Search a Food Item of a
Restaurant");


                            System.out.print("Enter Restaurant RID: ");
                            String rid4 = sc.next();
                            System.out.print("Enter Food Item fid: ");
                            String fid4 = sc.next();



                            FoodItem ff =
fc.getRestaurant(rid4).getFoodItem(fid4);




                            if(ff != null)
                            {
                                ff.showInfo();
                            }
```

```
                                              }
                                  catch (InputMismatchException ime17)
                   {
                                                      ime17.printStackTrace();

System.out.println("InputMismatchException occured");
                              }
                                              System.out.println("#####################");
                                              break;
                                      case 5:

                                              System.out.println("#####################");
                                              System.out.println("Going Back ....");
                                              System.out.println("#####################");
                                              break;

                                      default:

                                              System.out.println("#####################");
                                              System.out.println("Invalid Option");
                                              System.out.println("#####################");
                                              break;

                              }

                              System.out.println("********************");
                              break;
                               }
                                catch (InputMismatchException ime4)
                {
```

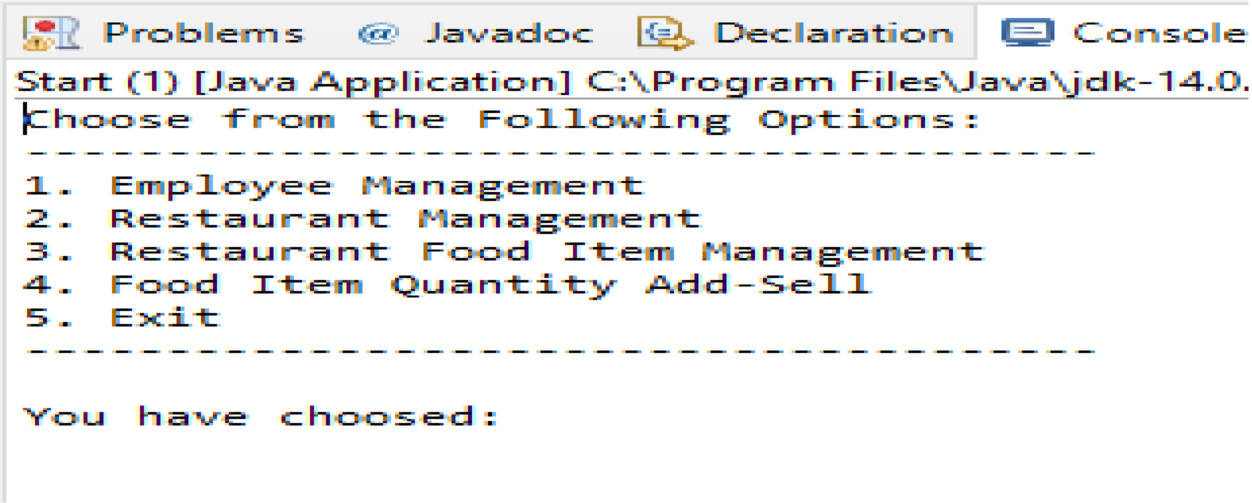
```
                                                          ime4.printStackTrace();

System.out.println("InputMismatchException occured");
                             }

                         case 4:
                           try{

                                 System.out.println("********************");
                                 System.out.println("Food Item  Add or Sell");



                                 System.out.println("What do you want to do?\n");
                                 System.out.println("--------------------------------------");
                                 System.out.println("1. Add Amount of Food");
                                 System.out.println("2. Sell Amount of Food");
                                 System.out.println("3. Show Add/Sell History");
                                 System.out.println("4. Go Back");
                                 System.out.println("--------------------------------------\n");

                                 System.out.print("Enter your option: ");
                                 int input4 = sc.nextInt();

                                 switch(input4)
                                 {
                                         case 1:

                                                 System.out.println("#####################");
                                                 System.out.println("Add Amount Of Food \n");

                                                 System.out.print("Enter Restaurant RID: ");
                                                 String rid1 = sc.next();
```

```
                            System.out.print("Enter Food Item fid: ");
                            String fid1 = sc.next();
                            System.out.print("Enter What Amount of Food
Will Add: ");
                            int amount1 = sc.nextInt();

                            if(amount1>0)
                            {
                                try{
    fc.getRestaurant(rid1).getFoodItem(fid1).addQuantity(amount1);


                                frwd.writeInFile("Add Quantity    : "+
amount1+" add with this fid "+ fid1);
                            }
                            catch(NullPointerException npe)
                            {
                                npe.printStackTrace();
                                if(fc.getRestaurant(rid1)==null)
                                {System.out.println(" Rid not found");}
                            else
                            {System.out.println(" Fid not found");
                        }
                            }
                            }

                            System.out.println("####################");
                            break;

                        case 2:

                            System.out.println("####################");
                            System.out.println("Sell Food");


                            System.out.print("Enter Restaurant RID: ");
                            String rid2 = sc.next();
                            System.out.print("Enter Food Item FID: ");
                            String fid2 = sc.next();
                            System.out.print("Enter What Amount You Want
To Sell: ");
                            int amount2 = sc.nextInt();



                            try{
                            if(amount2>0 && amount2 <=
fc.getRestaurant(rid2).getFoodItem(fid2).getAvailableQuantity())
                            {

    fc.getRestaurant(rid2).getFoodItem(fid2).sellQuantity(amount2);




                                frwd.writeInFile("Sell Quantity     : "+
amount2+" sell from this fid "+ fid2);
                            }
                            }
                            catch(NullPointerException npe)
                            {
                                npe.printStackTrace();
                                if(fc.getRestaurant(rid2)==null)
                                {System.out.println(" Rid not found");}
                            else
                            {System.out.println(" Fid not found");
                        }
                            }
```

```
                                     System.out.println("######################");
                                     break;



                               case 3:


                                 try{
                                     System.out.println("######################");
                                     System.out.println("Show add sell History");
                                     frwd.readFromFile();


                                      }
                           catch (InputMismatchException ime18)
                {
                                           ime18.printStackTrace();

System.out.println("InputMismatchException occured");
                         }
                                     System.out.println("######################");
                                     break;

                               case 4:


                                     System.out.println("######################");
                                     System.out.println("Going Back..");
                                     System.out.println("######################");
                                     break;

                               default:
```

```
                        System.out.println("#####################");
                        System.out.println("Invalid Option");
                        System.out.println("#####################");
                        break;
                }

                System.out.println("********************");
                break;
                }
                 catch (InputMismatchException ime5)
        {
                                    ime5.printStackTrace();

System.out.println("InputMismatchException occured");
             }

            case 5:
              try{

                System.out.println("********************");
                System.out.println("Exit");
                choice = false;
                System.out.println("********************");
                break;
                 }
                  catch (InputMismatchException ime6)
        {
                                    ime6.printStackTrace();

System.out.println("InputMismatchException occured");
              }
            default:
```

```
                         System.out.println("********************");
                         System.out.println("Invalid Option");
                         System.out.println("********************");
                         break;


                 }


              }



          }
          catch (InputMismatchException ime1)
             {
                                        ime1.printStackTrace();

     System.out.println("InputMismatchException occured");
                    }
                     catch (Exception e)
             {
                                        e.printStackTrace();
                                        System.out.println("Here Exception
occured");
                    }





}


}
```



```
Start (1) [Java Application] C:\Program Files\Java\jdk-14.0.
Choose from the Following Options:
-----------------------------------------
1. Employee Management
2. Restaurant Management
3. Restaurant Food Item Management
4. Food Item Quantity Add-Sell
5. Exit
-----------------------------------------

You have choosed:
```