**U. Mahfoos Ahamed**

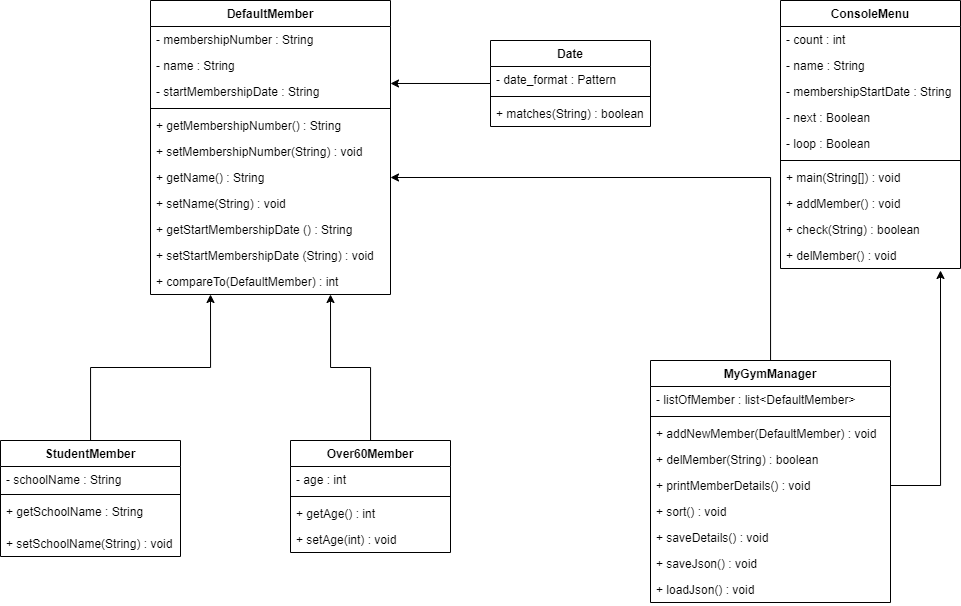
**W1790335**

**20191191**

**Programming Principles 2**

**YouTube Video Link: - https://youtu.be/\_1kwfYT6v\_k**

**Class Diagram**



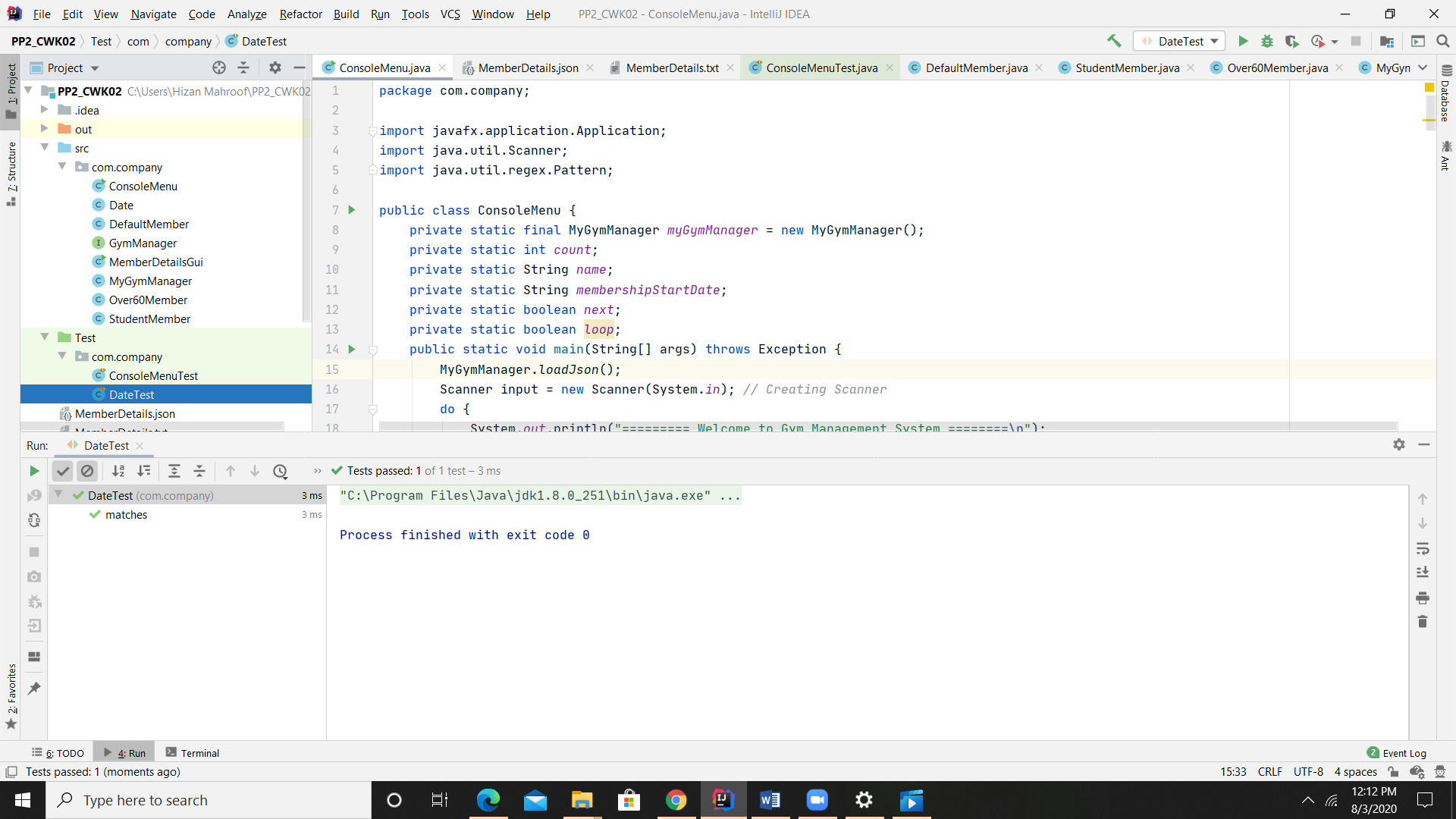
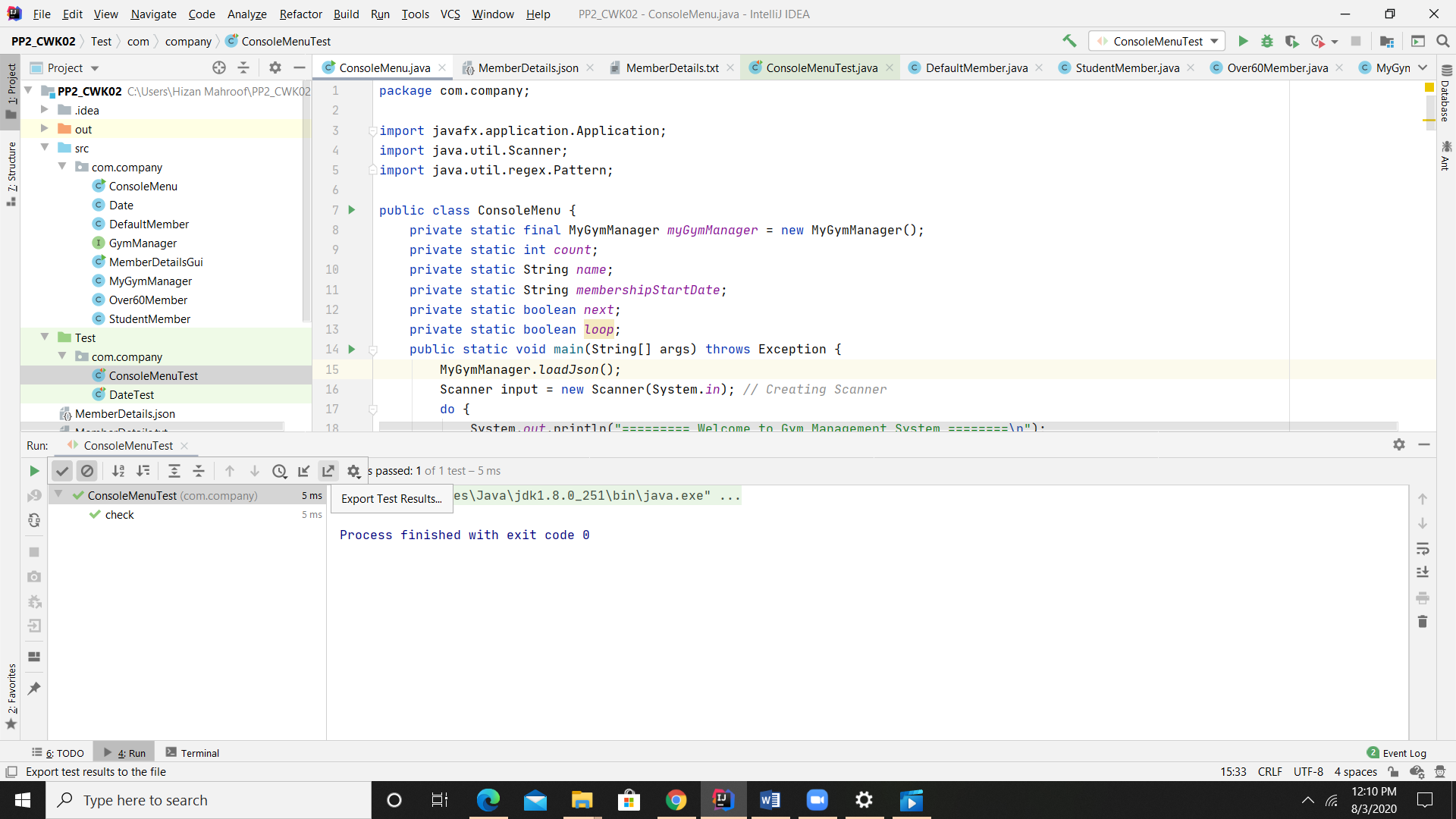
**Junit Codes**

**ConsoleMenuTest.java**

package com.company;  
  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class ConsoleMenuTest {  
  
  
 @Test  
 public void check() {  
 *assertTrue*(ConsoleMenu.*check*("Ahamed"));  
 *assertFalse*(ConsoleMenu.*check*("123"));  
 }  
}

**DateTest.java**

package com.company;  
  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class DateTest {  
  
 @Test  
 public void matches() {  
 *assertTrue*(new Date().matches("01-01-2020"));  
 *assertFalse*(new Date().matches("date"));  
 }  
}



**ConsoleMenu.java**

package com.company;  
  
import javafx.application.Application;  
import java.util.Scanner;  
import java.util.regex.Pattern;  
  
public class ConsoleMenu {  
 private static final MyGymManager *myGymManager* = new MyGymManager();  
 private static int *count*;  
 private static String *name*;  
 private static String *membershipStartDate*;  
 private static boolean *next*;  
 private static boolean *loop*;  
 public static void main(String[] args) throws Exception {  
 MyGymManager.*loadJson*();  
 Scanner input = new Scanner(System.*in*); *// Creating Scanner* do {  
 System.*out*.println("========= Welcome to Gym Management System ========\n");  
 System.*out*.println("\t\t\tEnter '1' for Add New Member: ");  
 System.*out*.println("\t\t\tEnter '2' for Delete Member Details : ");  
 System.*out*.println("\t\t\tEnter '3' for Display Member Details: ");  
 System.*out*.println("\t\t\tEnter '4' for Save the Member Details into text file: ");  
 System.*out*.println("\t\t\tEnter '5' for sorting the name ");  
 System.*out*.println("\t\t\tEnter '6' for Open Table view");  
 System.*out*.println("\t\t\tEnter '7' for Exit the System");  
 int option = input.nextInt();  
  
 switch (option) {  
 case 1:  
 *addMember*();  
 break;  
 case 2:  
 *delMember*();  
 break;  
 case 3:  
 *myGymManager*.printMemberDetails(); *// Print the Member Details in Console* break;  
 case 4:  
 *myGymManager*.saveDetails(); *// Save the Details into File* break;  
 case 5:  
 *myGymManager*.sort(); *// For Sort the name in Ascending Order* break;  
 case 6:  
 Application.*launch*(MemberDetailsGui.class); *// For gui* break;  
 case 7:  
 System.*exit*(0); *// System Exit* default:  
 System.*out*.println("\t\t\tInvalid Selection");  
 }  
  
 do {  
 System.*out*.print("\t\t\twhat u need ?\n\t\t\tQ = Quit\n\t\t\tC = Continue\n\t\t\tEnter you are choice : ");  
 String choice = input.next();  
 if (choice.equals("Q") || choice.equals("q")) {  
 *next* = false;  
 *loop* = false;  
 } else if (choice.equals("C") || choice.equals("c")) {  
 *next* = true;  
 *loop* = false;  
 }else {  
 *loop* = true;  
 System.*out*.println("\t\t\tPlease enter valid input");  
 }  
 }while (*loop*);  
  
  
 }while (*next*);  
 }  
  
 public static void addMember() {  
 Scanner usrInput = new Scanner(System.*in*); *// Creating Scanner* if (*count* < 100) {  
 System.*out*.print("\t\t\tEnter the Membership No: ");  
 String membershipNumber = usrInput.next();  
 boolean bool = true;  
  
 while (bool) {  
 System.*out*.print("\n\t\t\tEnter the Name: ");  
 *name* = usrInput.next();  
 bool = *check*(*name*);  
 if (!bool) {  
 System.*out*.println("\t\t\tPlease Enter Valid String");  
 bool = true;  
 } else {  
 bool = false;  
 }  
 }  
 bool = true;  
 Date date = new Date();  
 while (bool){  
 System.*out*.print("\n\t\t\tEnter the Date: following format DD-MM-YYYY :");  
 *membershipStartDate* = usrInput.next();  
 bool =date.matches(*membershipStartDate*);  
 if (!bool) {  
 System.*out*.println("\t\t\tPlease Enter Valid Date");  
 bool = true;  
 } else {  
 bool = false;  
 }  
 }  
 System.*out*.print("\n\t\t\tEnter the type of Membership ('D'- Default Member 'S' - Student Number 'O' - Over 60 Member) : ");  
 String option = usrInput.next();  
 DefaultMember defMember = null;  
 switch (option) {  
 case "D":  
 case "d":  
 defMember = new DefaultMember(membershipNumber, *name*, *membershipStartDate*);  
 *myGymManager*.addNewMember(defMember);  
 *count*++;  
 break;  
 case "S":  
 case "s":  
 System.*out*.print("\n\t\t\tEnter the School Name: ");  
 String schoolName = usrInput.next();  
 defMember = new StudentMember(membershipNumber, *name*, *membershipStartDate*, schoolName);  
 *myGymManager*.addNewMember(defMember);  
 *count*++;  
 break;  
 case "O":  
 case "o":  
 try {  
 int age = 0;  
 while (age < 60) {  
 System.*out*.print("\n\t\t\tEnter the Age: ");  
 age = usrInput.nextInt();  
 defMember = new Over60Member(membershipNumber, *name*, *membershipStartDate*, age);  
 }  
 *myGymManager*.addNewMember(defMember);  
 *count*++;  
 } catch (Exception e) {  
 System.*out*.println("\t\t\tPls enter integer value");  
 System.*out*.print("\n\t\t\tEnter the Age: \n");  
 }  
 break;  
 default:  
 System.*out*.print("\t\t\tInvalid Option Select");  
 break;  
 }  
  
 }  
 }  
  
  
 public static boolean check(String str) {  
 return Pattern.*matches*("[A-Za-z]+", str); *// Checking the name* }  
  
 public static void delMember() {  
 Scanner usrInput = new Scanner(System.*in*); *// Creating Scanner* System.*out*.print("\t\t\tEnter the Membership Number : ");  
 String membershipNo = usrInput.next();  
 boolean output = *myGymManager*.delMember(membershipNo);  
 if (output) {  
 *count*--;  
 }  
  
 }  
  
}

**DefaultMember.java**

package com.company;  
  
public class DefaultMember implements Comparable<DefaultMember>{  
 private String membershipNumber;  
 private String name;  
 private String startMembershipDate;  
  
 public DefaultMember(String membershipNumber, String name, String startMembershipDate) {  
 this.membershipNumber = membershipNumber;  
 this.name = name;  
 setStartMembershipDate(startMembershipDate);  
 }  
  
 public String getMembershipNumber() {  
 return membershipNumber;  
 }  
  
 public void setMembershipNumber(String membershipNo) {  
 this.membershipNumber = membershipNo;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String membName) {  
 this.name = membName;  
 }  
  
 public String getStartMembershipDate() {  
 return startMembershipDate;  
 }  
  
 public void setStartMembershipDate(String date) {  
 Date nnn = new Date();  
 boolean test =nnn.matches(date);  
 if(test == true) {  
 this.startMembershipDate = date;  
 }  
 }

@Override  
 public int compareTo(DefaultMember o) {  
 return this.name.compareTo(o.getName());  
 }  
}

**StudentMember.java**

package com.company;  
  
public class StudentMember extends DefaultMember {  
 private String schoolName;  
  
 public StudentMember(String membershipNumber, String name, String startMembershipDate, String schoolName) {  
 super(membershipNumber, name, startMembershipDate);  
 this.schoolName = schoolName;  
 }  
  
 public String getSchoolName(){  
 return schoolName;  
 }  
  
 public void setSchoolName(String schoolName){  
 this.schoolName = schoolName;  
 }  
}

**Over60Member.java**

package com.company;  
  
public class StudentMember extends DefaultMember {  
 private String schoolName;  
  
 public StudentMember(String membershipNumber, String name, String startMembershipDate, String schoolName) {  
 super(membershipNumber, name, startMembershipDate);  
 this.schoolName = schoolName;  
 }  
  
 public String getSchoolName(){  
 return schoolName;  
 }  
  
 public void setSchoolName(String schoolName){  
 this.schoolName = schoolName;  
 }  
}

**Date.java**

package com.company;  
  
import java.util.regex.Pattern;  
  
public class Date {  
 private static Pattern *date\_format* = Pattern.*compile*(  
 "^(29-02-(2000|2400|2800|(19|2[0-9](0[48]|[2468][048]|[13579][26]))))$"  
 + "|^((0[1-9]|1[0-9]|2[0-8])-02-((19|2[0-9])[0-9]{2}))$"  
 + "|^((0[1-9]|[12][0-9]|3[01])-(0[13578]|10|12)-((19|2[0-9])[0-9]{2}))$"  
 + "|^((0[1-9]|[12][0-9]|30)-(0[469]|11)-((19|2[0-9])[0-9]{2}))$");  
  
 public boolean matches(String date) {  
 return *date\_format*.matcher(date).matches();  
 }  
  
}

**GymManager.java**

package com.company;  
  
  
import java.io.IOException;  
  
public interface GymManager {  
 void addNewMember(DefaultMember member);  
 boolean delMember(String membershipNo);  
 void printMemberDetails();  
 void sort();  
 void saveDetails() throws IOException;  
}

**MyGymManager.java**

package com.company;  
  
  
import org.json.JSONArray;  
import org.json.JSONObject;  
import org.json.JSONTokener;  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
  
public class MyGymManager implements GymManager {  
  
 public static List<DefaultMember> *lisOfMember* = new ArrayList<DefaultMember>();  
  
 @Override  
 public void addNewMember(DefaultMember defMember) {  
 if (*lisOfMember*.size() < 100) {  
 *lisOfMember*.add(defMember);  
 System.*out*.println("\n\t\t\tNo of Registered Member: " + *lisOfMember*.size());  
 System.*out*.println("\t\t\tNumber of Space Available: " + (100 - *lisOfMember*.size()));  
 } else {  
 System.*out*.println(" No space Available");  
 }  
  
 }  
  
  
 @Override  
 public boolean delMember(String membershipNo) {  
 boolean bool = false;  
 for (DefaultMember defMember : *lisOfMember*) {  
 if (defMember.getMembershipNumber().equals(membershipNo)) {  
 bool = true;  
 *lisOfMember*.remove(defMember);  
 System.*out*.println("Member with the membership no " + membershipNo + " Successfully removed");  
 System.*out*.println("No of Registered Member " + *lisOfMember*.size());  
 System.*out*.println("No of Free Space " + (100 - *lisOfMember*.size()));  
 if (defMember instanceof StudentMember) {  
 System.*out*.println("Member type is : Student Member");  
 } else if (defMember instanceof Over60Member) {  
 System.*out*.println("Member type is : Over60Member");  
 } else {  
 System.*out*.println("Member type is : DefaultMember");  
 }  
 break;  
 }  
  
 }  
 if (!bool) {  
 System.*out*.println("Not found");  
  
 }  
 return bool;  
 }  
  
 @Override  
 public void printMemberDetails() {  
 for (DefaultMember defMember : *lisOfMember*) {  
 System.*out*.print("\t\t\tMembership Number: " + defMember.getMembershipNumber() + " ");  
 if (defMember instanceof StudentMember) {  
 System.*out*.print("\tMember type is : StudentMember");  
 } else if (defMember instanceof Over60Member) {  
 System.*out*.print("\tMember type is : Over60Member");  
 } else {  
 System.*out*.print("\tMember type is : DefaultMember");  
 }  
 System.*out*.print("\tName is : " + defMember.getName() + " ");  
 System.*out*.println("\tMembership start is : " + defMember.getStartMembershipDate());  
 }  
 }  
  
 @Override  
 public void sort() {  
 Collections.*sort*(*lisOfMember*);  
 System.*out*.println("Successfully Sorted");  
  
 }  
  
 @Override  
 public void saveDetails() {  
 File file = new File("MemberDetails.txt");  
 try {  
  
 FileWriter out = new FileWriter(file, true);  
 for (DefaultMember defMember : *lisOfMember*) {  
 out.write("MembershipNumber: " + defMember.getMembershipNumber() + " ");  
 if (defMember instanceof StudentMember) {  
 out.write("\tMember type is : StudentMember");  
 } else if (defMember instanceof Over60Member) {  
 out.write("\tMember type is : Over60Member");  
 } else {  
 out.write("\tMember type is : DefaultMember");  
 }  
 out.write("\tName is : " + defMember.getName());  
 out.write("\tMembership start is : " + defMember.getStartMembershipDate() + "\n");  
 }  
 saveJson();  
 System.*out*.println("\t\t\tMember Details saved to file Successfully");  
 out.close();  
 } catch (Exception e) {  
 System.*out*.println("\t\t\tNo Data Saved");  
 }  
  
 }  
  
 public void saveJson(){  
 File file = new File("MemberDetails.json");  
 try {  
 FileWriter out = new FileWriter(file);  
 JSONArray jsonArray = new JSONArray();  
 for (DefaultMember defMember : *lisOfMember*) {  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("MembershipNumber" ,defMember.getMembershipNumber());  
 if (defMember instanceof StudentMember) {  
 jsonObject.put("Member\_type\_is","StudentMember");  
 jsonObject.put("School\_Name",((StudentMember) defMember).getSchoolName());  
 } else if (defMember instanceof Over60Member) {  
 jsonObject.put("Member\_type\_is","Over60Member");  
 jsonObject.put("Member\_Age",((Over60Member) defMember).getAge());  
 } else {  
 jsonObject.put("Member\_type\_is","DefaultMember");  
 }  
 jsonObject.put("Name\_is",defMember.getName());  
 jsonObject.put("Membership\_start\_is" , defMember.getStartMembershipDate());  
 jsonArray.put(jsonObject);  
 }  
 out.write(String.*valueOf*(jsonArray));  
 out.flush();  
 out.close();  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 public static void loadJson() throws Exception{  
 try {  
 InputStream inputStr = new FileInputStream("MemberDetails.json");  
 JSONTokener tokenerJson = new JSONTokener(inputStr);  
 JSONArray arrayJson = new JSONArray(tokenerJson);  
  
 for (int i =0; i <arrayJson.length();i++){  
 JSONObject jsonObject = arrayJson.getJSONObject(i);  
 String str = jsonObject.getString("MembershipNumber");  
 String strName = jsonObject.getString("Name\_is");  
 String type = jsonObject.getString("Member\_type\_is");  
 String strDate = jsonObject.getString("Membership\_start\_is");  
 switch (type){  
 case "DefaultMember":  
 *lisOfMember*.add(new DefaultMember(str,strName,strDate));  
 break;  
 case "StudentMember":  
 String school = jsonObject.getString("School\_Name");  
 *lisOfMember*.add(new StudentMember(str,strName,strDate,school));  
 break;  
 case "Over60Member":  
 int over60 = jsonObject.getInt("Member\_Age");  
 *lisOfMember*.add(new Over60Member(str,strName,strDate,over60));  
 break;  
  
 }  
  
 }  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
  
}

**MemberDetailsGui.java**

package com.company;  
  
import javafx.application.Application;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.scene.Scene;  
import javafx.scene.control.Label;  
import javafx.scene.control.SelectionMode;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
  
public class MemberDetailsGui extends Application {  
  
 @Override public void start(Stage stage) {  
  
 Label lbl = new Label("Members Details"); *// Label For Table Name* TableView <DefaultMember> tableMember = new TableView<DefaultMember>();  
 final ObservableList<DefaultMember> observer = FXCollections.*observableArrayList*(MyGymManager.*lisOfMember*);  
  
 TableColumn memberId = new TableColumn("Membership Number"); *// Creating Membership Number Column* memberId.setCellValueFactory(new PropertyValueFactory("membershipNumber"));  
  
 TableColumn memberName = new TableColumn("Member Name"); *// Creating Member Name Column* memberName.setCellValueFactory(new PropertyValueFactory("name"));  
  
 TableColumn memberType = new TableColumn("Member Enrolled Date"); *// Creating Member Enrolled Date Column* memberType.setCellValueFactory(new PropertyValueFactory("startMembershipDate"));  
  
 TableColumn memberSchool = new TableColumn("School Name"); *// Creating School Name Column* memberSchool.setCellValueFactory(new PropertyValueFactory("schoolName"));

TableColumn memberAge = new TableColumn("Member Age"); *// Creating Member Age Column* memberAge.setCellValueFactory(new PropertyValueFactory("age"));  
  
 tableMember.setItems(observer);  
 tableMember.getSelectionModel().setSelectionMode(SelectionMode.*MULTIPLE*);  
 tableMember.getColumns().addAll(memberId,memberName,memberType,memberSchool,memberAge);  
  
 tableMember.setMaxSize(600,400);  
 VBox tblBox = new VBox();  
 tblBox.setSpacing(5);  
 tblBox.setPadding(new Insets(10,50,50,50));  
 tblBox.getChildren().addAll(lbl,tableMember);  
  
 Scene tblScene = new Scene(tblBox,900,500);  
 stage.setTitle("Members Details");  
 stage.setScene(tblScene);  
 stage.sizeToScene();  
 stage.show();  
  
 }  
  
 public static void main(String[] args) {  
 Application.*launch*(args);  
 }  
}