**ICT 131 : Introductory Programming and Object- Oriented Concepts Using Java**

**Tutor-Marked Assignment**

**Question 1(a):**

import java.util.Scanner;

public class Question1\_a

{

public static void main(String args[])

{

System.out.println("Student name: Wang Yuet Hua Jennifer");

System.out.println("Student number: B1571734");

int output;

Scanner input1 = new Scanner(System.in);

int num1 = input1.nextInt();

int x1 = num1 / 100;

int x2 = (num1 % 100)/10;

int x3 = num1 % 10;

System.out.println(x1 + " " + x2 + " " + x3);

Scanner input2 = new Scanner(System.in);

int num2 = input2.nextInt();

int y1 = num2 / 100;

int y2 = (num2 % 100)/10;

int y3 = num2 % 10;

System.out.println("- " + y1 + " " + y2 + " " + y3);

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

output = num1 - num2;

int o1 = output / 100;

int o2 = (output % 100)/10;

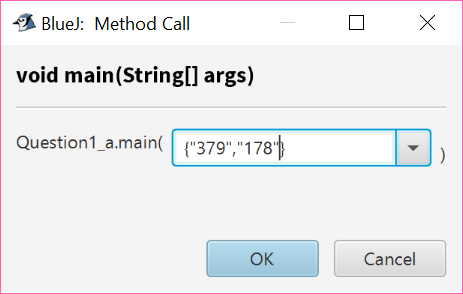
int o3 = output % 10;

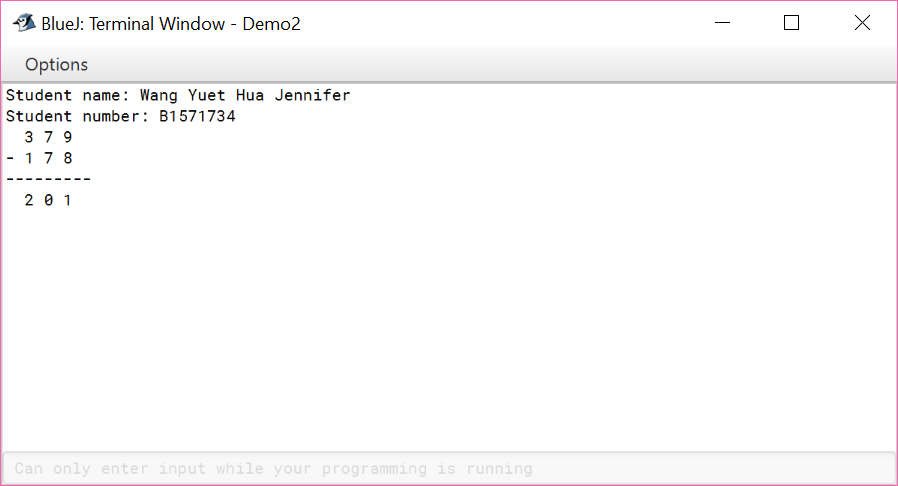
System.out.println(o1 + " " + o2 + " " + o3);

}

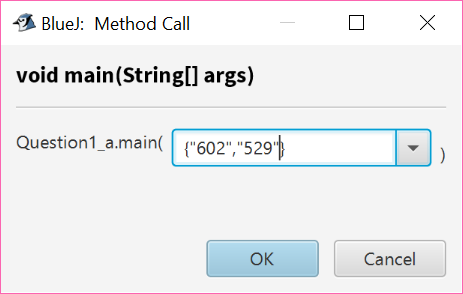
}

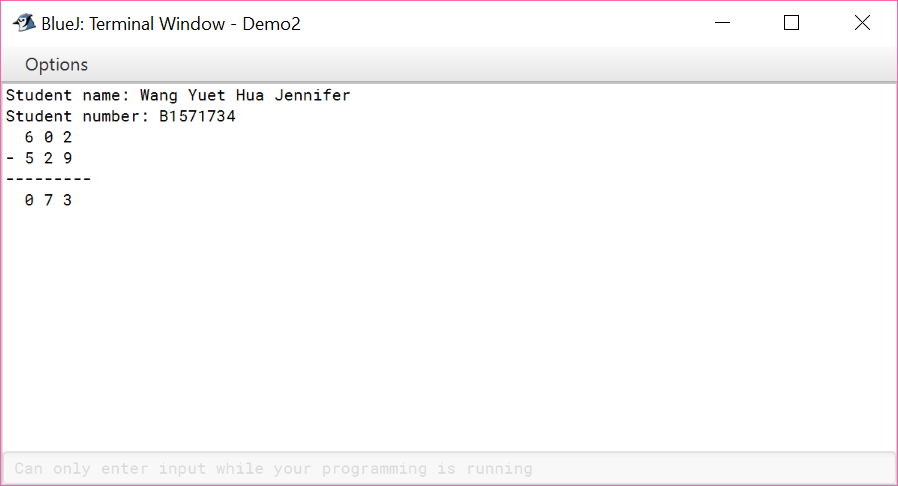
**Screenshot 1:**



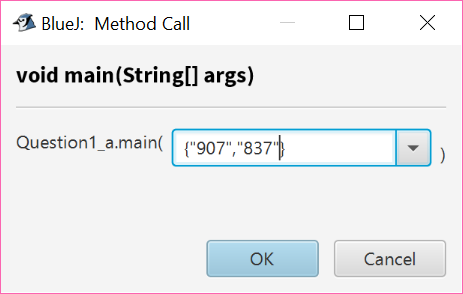


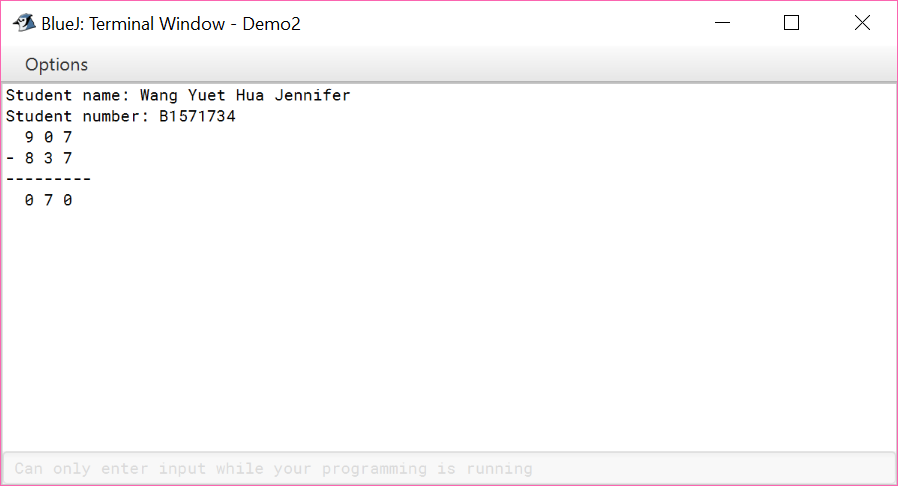
**Screenshot 2:**





**Screenshot 3:**





**Question 1(b):**

import java.util.Scanner;

public class Question1\_b

{

public static void main (String [] args)

{

Scanner input1 = new Scanner (System.in);

int choice = 0;

System.out.println("Choose the type of monthly pass");

System.out.println("1 - Primary Student");

System.out.println("2 - Secondary Student");

System.out.println("3 - Polytechnic Student");

System.out.println("4 - University Student");

System.out.println("5 - Full-Time National Serviceman");

System.out.println("6 - Adult");

System.out.println("7 - Senior Citizen");

System.out.println("8 - Person with Disability");

System.out.print("Your choice: ");

choice = input1.nextInt();

if (choice == 0)

{

System.out.println("Invalid type of person");

}

else

{

switch (choice)

{

case 1: case 2: case 3: case 4: case 5:

System.out.println("Enter the type of monthly pass");

System.out.println("B - Bus");

System.out.println("T- Train");

System.out.println("H - Hybrid (Bus and Train)");

Scanner input2 = new Scanner (System.in);

System.out.print("Your choice: ");

String mthpasschoice = input2.nextLine();

if (mthpasschoice.equalsIgnoreCase("B") && choice==1)

{

System.out.print("Cost of pass = $22.50");

}

if (mthpasschoice.equalsIgnoreCase("T") && choice==1)

{

System.out.print("Cost of pass = $20.00");

}

if (mthpasschoice.equalsIgnoreCase("H") && choice==1)

{

System.out.print("Cost of pass = $41.00");

}

if (mthpasschoice.equalsIgnoreCase("B") && choice==2)

{

System.out.print("Cost of pass = $27.50");

}

if (mthpasschoice.equalsIgnoreCase("T") && choice==2)

{

System.out.print("Cost of pass = $25.00");

}

if (mthpasschoice.equalsIgnoreCase("H") && choice==2)

{

System.out.print("Cost of pass = $51.00");

}

if (mthpasschoice.equalsIgnoreCase("B") && choice==3)

{

System.out.print("Cost of pass = $27.50");

}

if (mthpasschoice.equalsIgnoreCase("T") && choice==3)

{

System.out.print("Cost of pass = $25.00");

}

if (mthpasschoice.equalsIgnoreCase("H") && choice==3)

{

System.out.print("Cost of pass = $51.00");

}

if (mthpasschoice.equalsIgnoreCase("B") && choice==4)

{

System.out.print("Cost of pass = $52.00");

}

if (mthpasschoice.equalsIgnoreCase("T") && choice==4)

{

System.out.print("Cost of pass = $45.00");

}

if (mthpasschoice.equalsIgnoreCase("H") && choice==4)

{

System.out.print("Cost of pass = $85.00");

}

if (mthpasschoice.equalsIgnoreCase("B") && choice==5)

{

System.out.print("Cost of pass = $52.00");

}

if (mthpasschoice.equalsIgnoreCase("T") && choice==5)

{

System.out.print("Cost of pass = $45.00");

}

if (mthpasschoice.equalsIgnoreCase("H") && choice==5)

{

System.out.print("Cost of pass = $85.00");

}

else

{

System.out.print("Invalid type of pass");

}

}

if (choice==6)

{

System.out.print("Cost of pass = $120.00");

}

if (choice==7 || choice==8)

{

System.out.print("Cost of pass = $60.00");

}

}

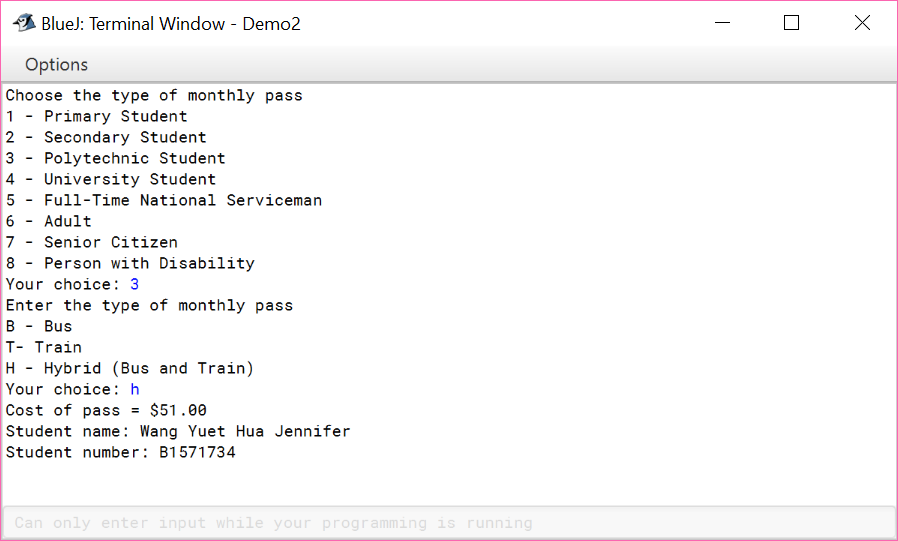
System.out.println("\nStudent name: Wang Yuet Hua Jennifer");

System.out.println("Student number: B1571734");

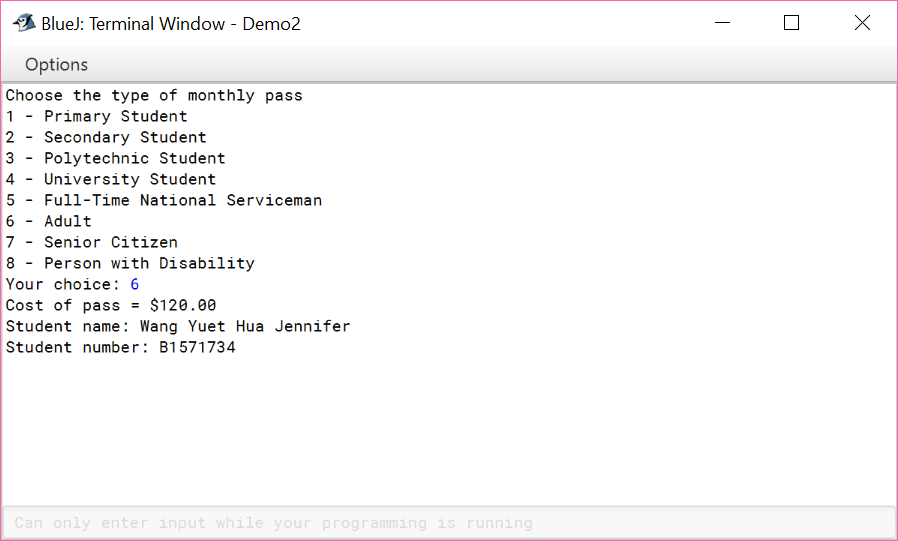
}

}

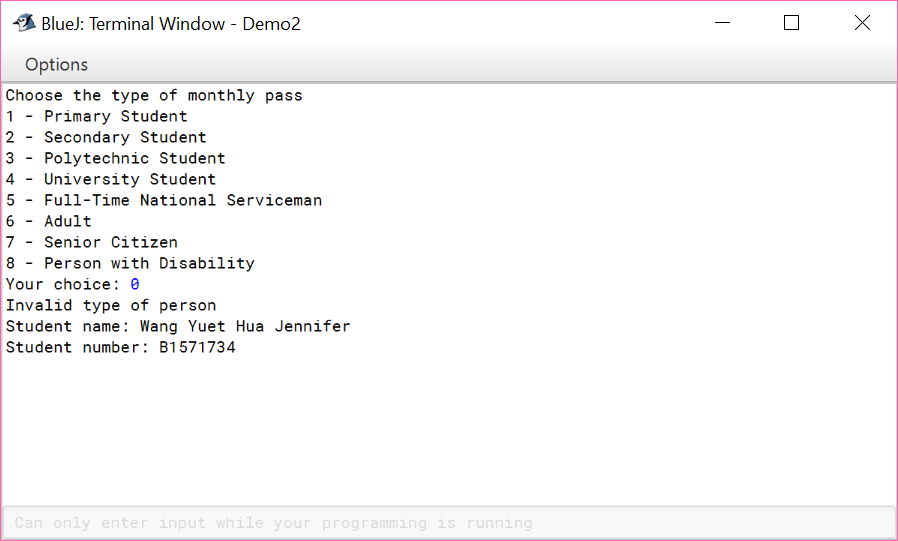
**Screenshot 1:**



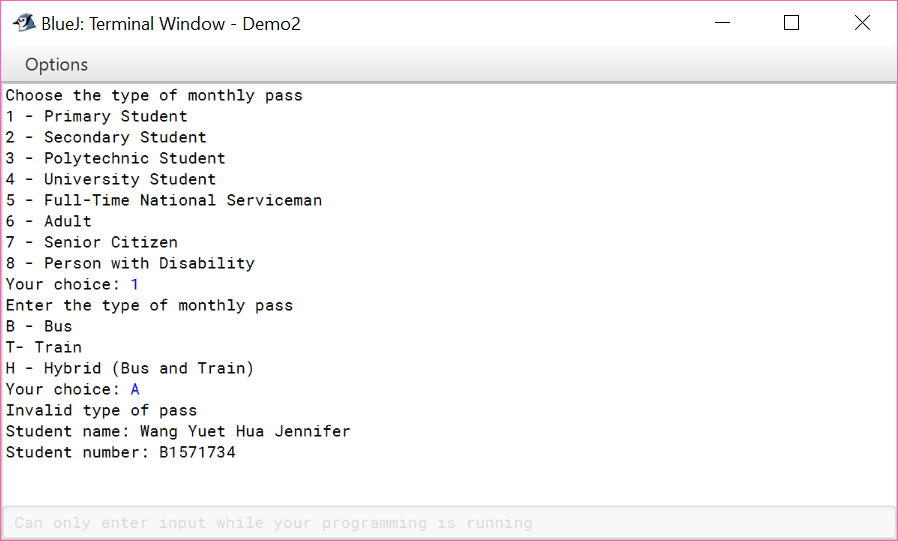
**Screenshot 2:**



**Screenshot 3:**



**Screenshot 4:**



**Question 2:**

import java.util.Scanner;

public class Question2

{

public static void main(String[] args)

{

Scanner console = new Scanner(System.in);

String s = "meringue albumen foulard eudaemonic narcolepsy elucubrate"

+ " vivisepulture pococurante cymotrichous malfeasance";

Scanner scanString;

scanString = new Scanner(s);

String word = "";

char spell=' ';

int length;

int start = 0;

int tries = 0;

int guess = 0;

do {

int index = (int) (Math.random()\*4); //cast this to get the 4

for(int n=0; n<=index; n++)

{

word = scanString.next();

}

scanString.close();

scanString = new Scanner(s);

length = word.length();

StringBuffer answer = new StringBuffer(word);

for (int n=0; n<answer.length(); n++)

{

answer.setCharAt(n, '-');

}

System.out.println("Spell this " + length + "-letter word in 5 tries");

System.out.println(answer);

while ((spell==' ' || spell=='n') && (tries<5)) {

System.out.print("Enter a letter: ");

char c = console.next().charAt(0);

tries++;

for(int n=0; n<answer.length(); n++)

{

if (word.charAt(n) == c)

{

answer.setCharAt(n,c);

guess = 1;

}

}

System.out.println(answer);

if ((tries<5) && (guess==1)) {

System.out.println("Do you want spell the word now? (y/n): ");

spell = console.next().charAt(0);

}

}

System.out.println("Spell the complete word: ");

String complete = console.next();

if (complete.equals(word))

System.out.println("You are correct!");

else

System.out.println("You are incorrect.");

System.out.println("The correct word is " + word);

System.out.print("Spell another word? (y/n): ");

char cont = console.next().charAt(0);

if (cont=='n') {

System.out.println("Thank you for playing Spell The Word!");

start = 0;

}

else  
 {

start = 1;

tries = 0;

guess = 0;

spell = ' ';

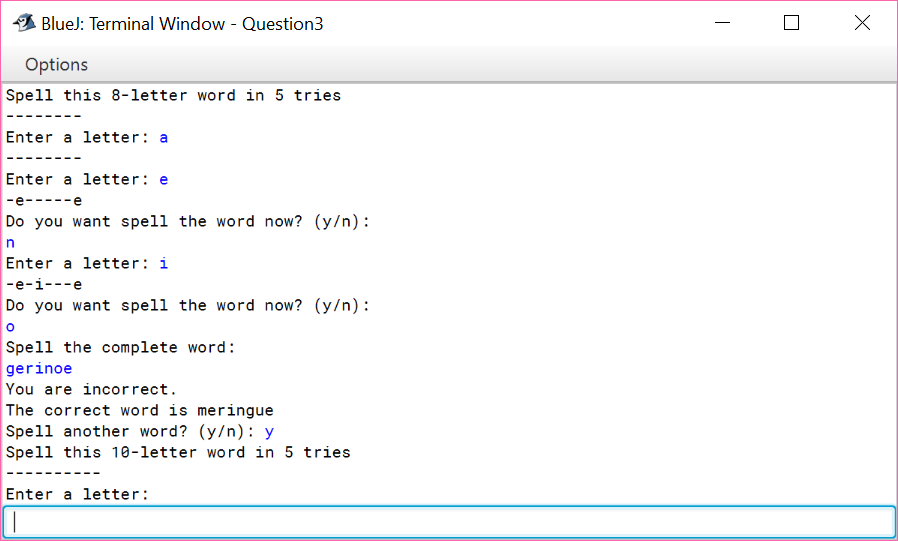
}

} while (start==1);

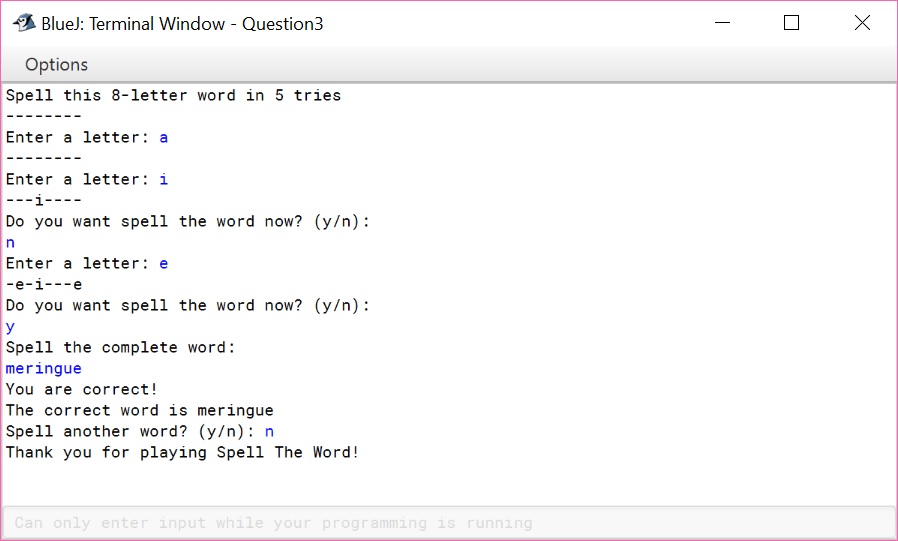
}

}

**Screenshot of run 1:**



**Screenshot of run 2:**



**Question 3:**

import java.util.Scanner;

import java.util.Arrays;

public class Question3

{

public static void main (String [] args)

{

char[] staffPositionCode = new char[2];

String[] StaffNames = new String[2];

int[] staffNumleaves = new int[2];

char[] positionCode = {'T', 'M', 'S', 'J'};

int[] numleaves = {52, 28, 21, 14};

int count = 0;

char code = ' ';

int option;

Scanner scanOption = new Scanner(System.in);

do {

displayMenu();

System.out.print("Enter option: ");

option = scanOption.nextInt();

switch(option) {

case 1:

if (count==2)

System.out.println("No more staff can be added.");

else {

scanOption.nextLine();

System.out.print("Enter name of staff to add: ");

String addedName = scanOption.nextLine();

System.out.print("Enter position code of staff to add (T, M, S or J): ");

code = scanOption.next().charAt(0);

count = addStaff(addedName, code, StaffNames, staffPositionCode, staffNumleaves, count, positionCode, numleaves);

}

break;

case 2:

scanOption.nextLine();

System.out.print("Enter name of staff to remove: ");

String removeName = scanOption.nextLine();

boolean isRemoved = removeStaff(removeName, StaffNames, staffPositionCode, staffNumleaves, count);

if (isRemoved)

count--;

break;

case 3:

scanOption.nextLine();

System.out.print("Enter name of staff taking leave: ");

String staff = scanOption.nextLine();

System.out.print("Enter number of days to apply leave for: ");

int numOfLeaves = scanOption.nextInt();

int checkLeaveApproved = takeLeave(staff, numOfLeaves, StaffNames, staffNumleaves, count);

if ((checkLeaveApproved>=0) && (checkLeaveApproved!=Integer.MAX\_VALUE))

System.out.println(staff + " has " + checkLeaveApproved + " day(s) of leave left");

else if (checkLeaveApproved<0) {

System.out.println("Applying " + numOfLeaves + "days will result in " + String.valueOf(checkLeaveApproved) + " days leave!");

System.out.println(staff + " can apply only " + (checkLeaveApproved+numOfLeaves) + " day(s) of leave");

}

break;

case 4:

listStaffWithLeave(StaffNames, staffNumleaves, count);

break;

default:

System.out.println("Closing application");

System.exit(0);

break;

}

} while(option!=0);

}

public static int search(String name, String[] staffNames, int count)

{

String lowerCase;

String upperCase;

for (int index=0; index<count; index++) {

lowerCase = staffNames[index].substring(0,1).toLowerCase() + staffNames[index].substring(1);

upperCase = staffNames[index].substring(0,1).toUpperCase() + staffNames[index].substring(1);

if ((name.equals(staffNames[index])) || (name.equals(upperCase)) || (name.equals(lowerCase)))

return index;

}

return -1;

}

public static int addStaff(String name, char positioncode, String[] staffNames, char[] staffPositionCode, int[] staffNumleaves, int count, char[] position, int[] numleaves)

{

if (search(name, staffNames, count) != -1) //name exists in staffNames

System.out.println(name + " is already a staff!");

else { //name doesn't exists in staffNames

staffNames[count] = name;

staffPositionCode[count] = positioncode;

for (int findCode=0; findCode<4; findCode++) {

if (positioncode == position[findCode]) {

staffNumleaves[count] = numleaves[findCode];

System.out.println("Successfully added " + name + " with " + numleaves[findCode] + " days of leave");

}

}

count++;

}

return count;

}

public static boolean removeStaff(String name, String[] staffNames, char[] staffPositionCode, int[] staffNumleaves, int count)

{

if (search(name, staffNames, count) == -1) { //name doesn't exists in staffNames

System.out.println("Can't remove " + name + ": Reason: Not a staff");

return false;

}

else {

for (int staffIndex=search(name, staffNames, count)+1; staffIndex<count; staffIndex++) {

staffNames[staffIndex-1] = staffNames[staffIndex];

staffPositionCode[staffIndex-1] = staffPositionCode[staffIndex];

staffNumleaves[staffIndex-1] = staffNumleaves[staffIndex];

}

System.out.println("Successfully removed: " + name);

return true;

}

}

public static int takeLeave(String name, int numDays, String[] staffNames, int[] staffNumleaves, int count)

{

if (search(name, staffNames, count) == -1) { //name doesn't exists in staffNames

System.out.println("Can't apply leave " + name + ": Reason: Not a staff");

return Integer.MAX\_VALUE;

}

else {

if (numDays > staffNumleaves[search(name, staffNames, count)]) {

System.out.println("Can't apply leave " + name + ": Reason: Insufficient leave");

return (staffNumleaves[search(name, staffNames, count)] - numDays); //return negative value

}

else {

staffNumleaves[search(name, staffNames, count)] = staffNumleaves[search(name, staffNames, count)] - numDays;

System.out.println("Successful leave application for " + name);

return staffNumleaves[search(name, staffNames, count)]; // return remaining leaves

}

}

}

public static void listStaffWithLeave(String[] staffNames, int[] staffNumleaves, int count)

{

System.out.println("List of Staff who can Take Leave");

for (int displayIndex=0; displayIndex<count; displayIndex++) {

System.out.printf("%-5s %5d days %n", staffNames[displayIndex], staffNumleaves[displayIndex]);

}

System.out.println("End of List");

}

public static void displayMenu()

{

System.out.println("Menu");

System.out.println("1. Add Staff");

System.out.println("2. Remove Staff");

System.out.println("3. Take Leave");

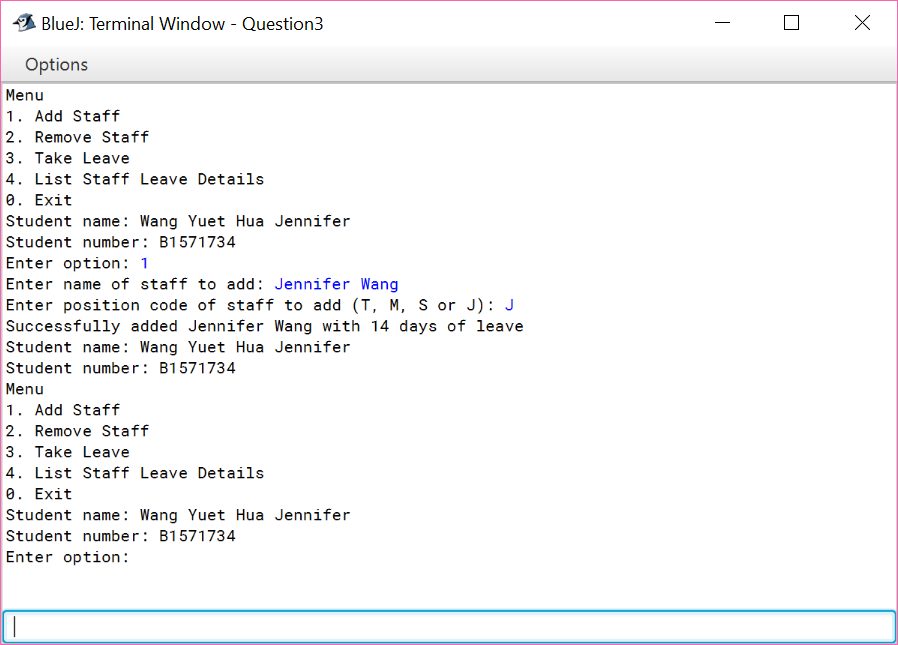
System.out.println("4. List Staff Leave Details");

System.out.println("0. Exit");

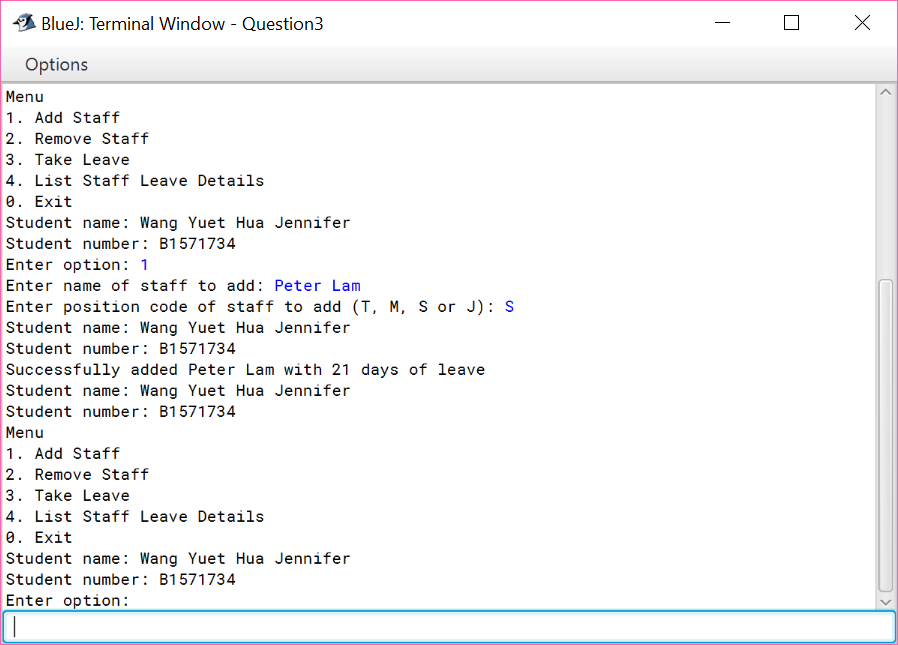
}

}

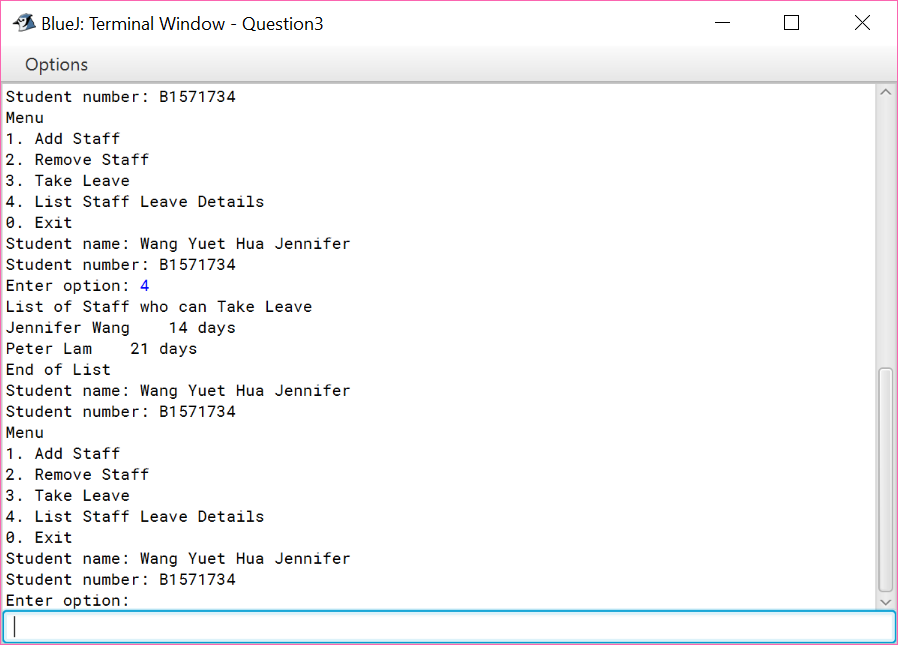
**Screenshot 1:**



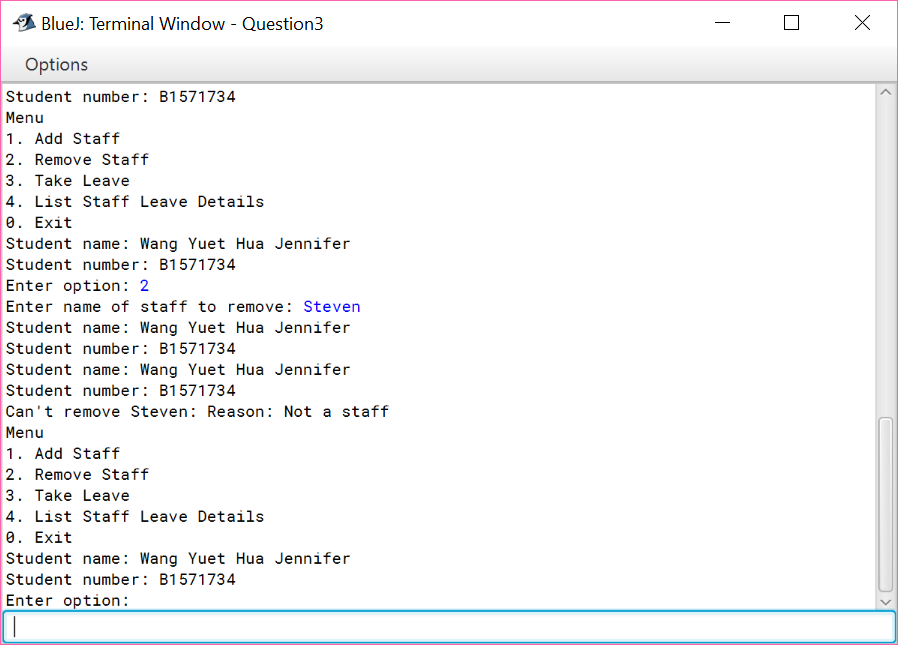
**Screenshot 2:**



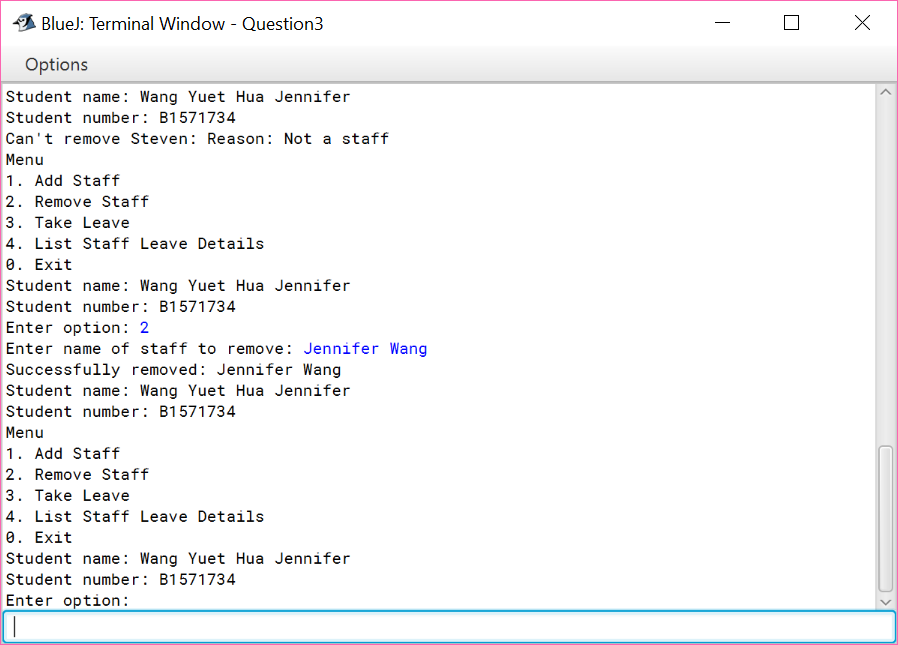
**Screenshot 3:**



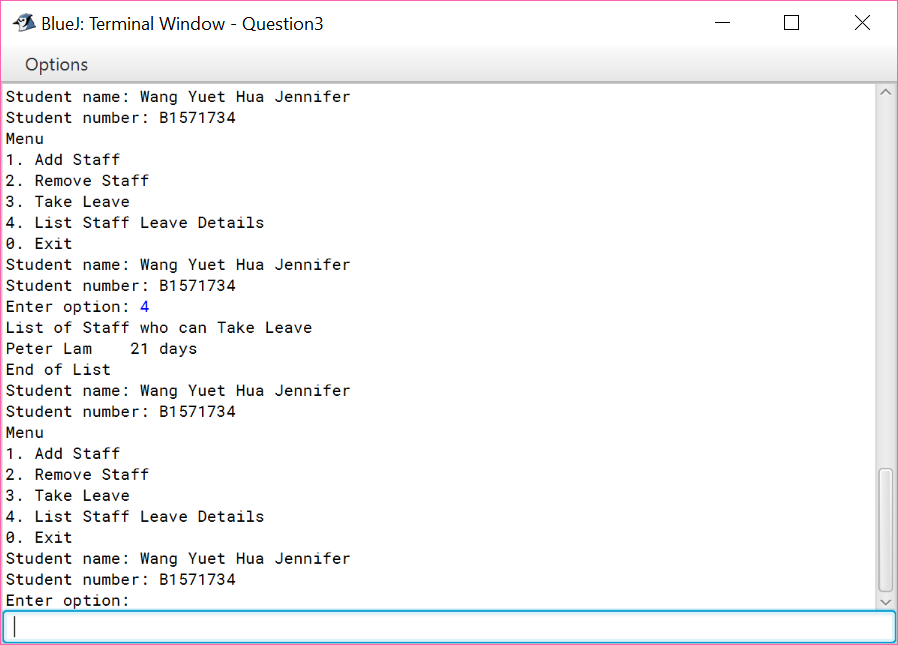
**Screenshot 4:**



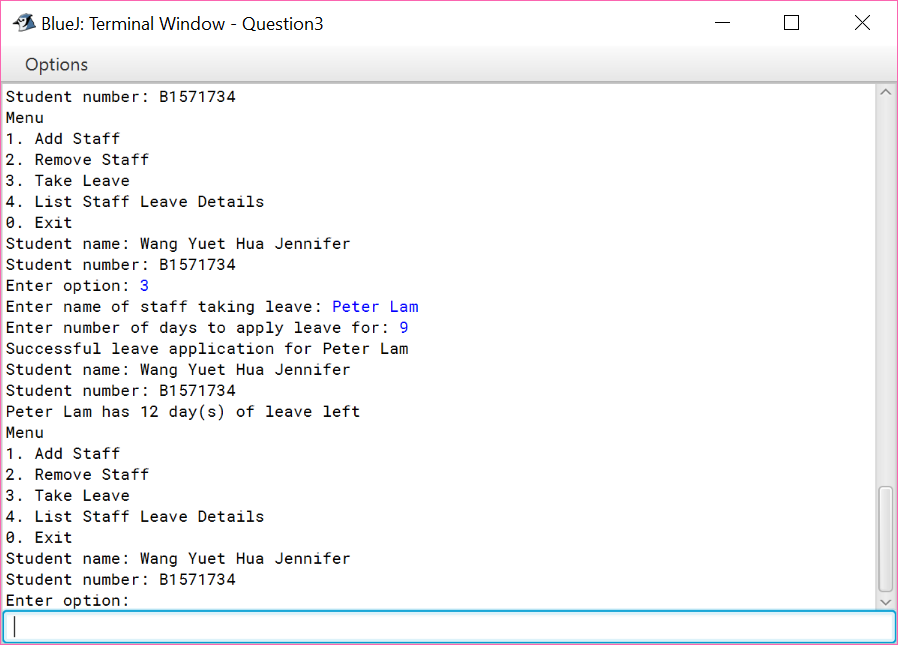
**Screenshot 5:**



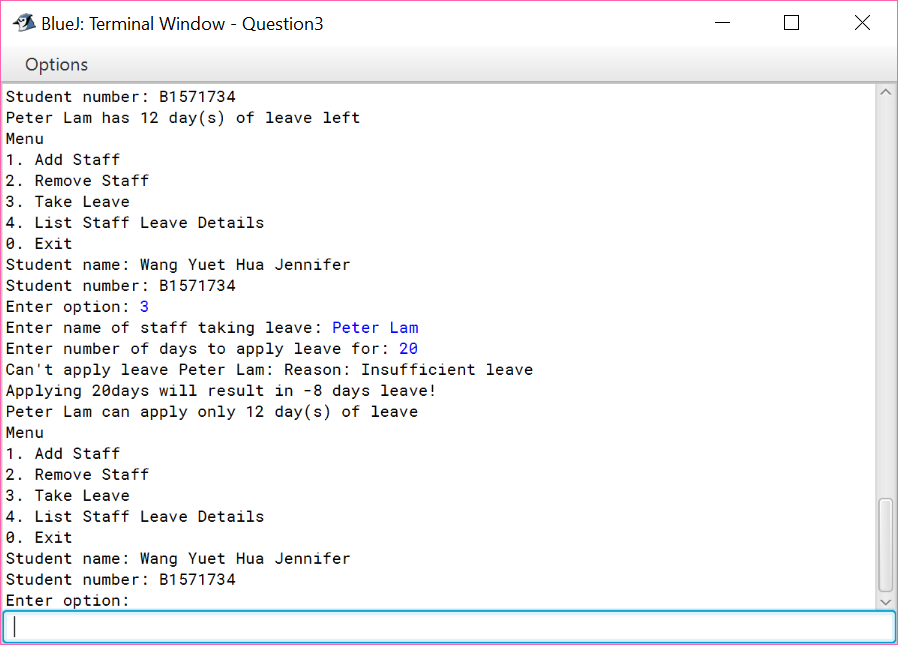
**Screenshot 6:**



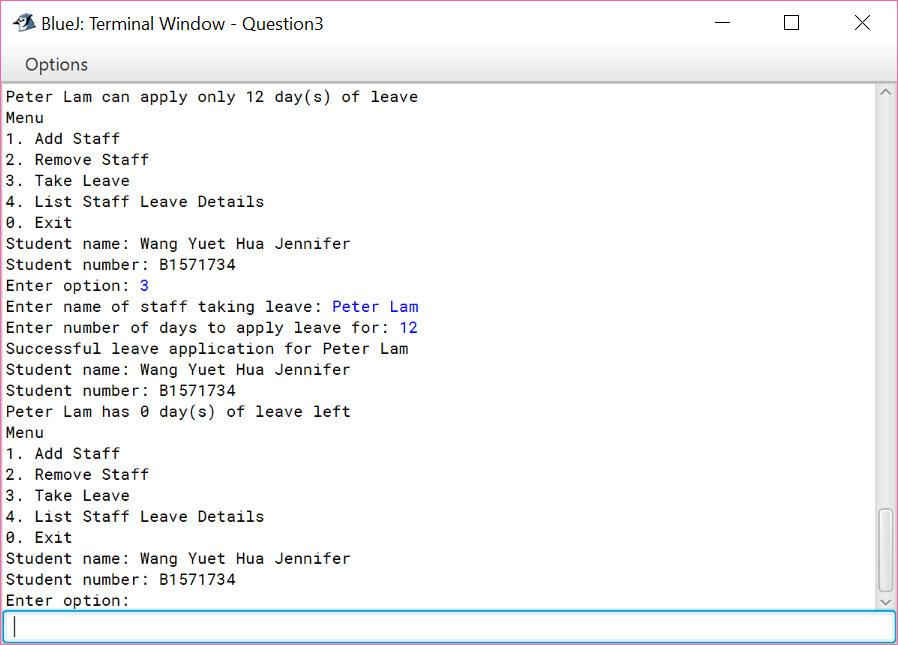
**Screenshot 7:**



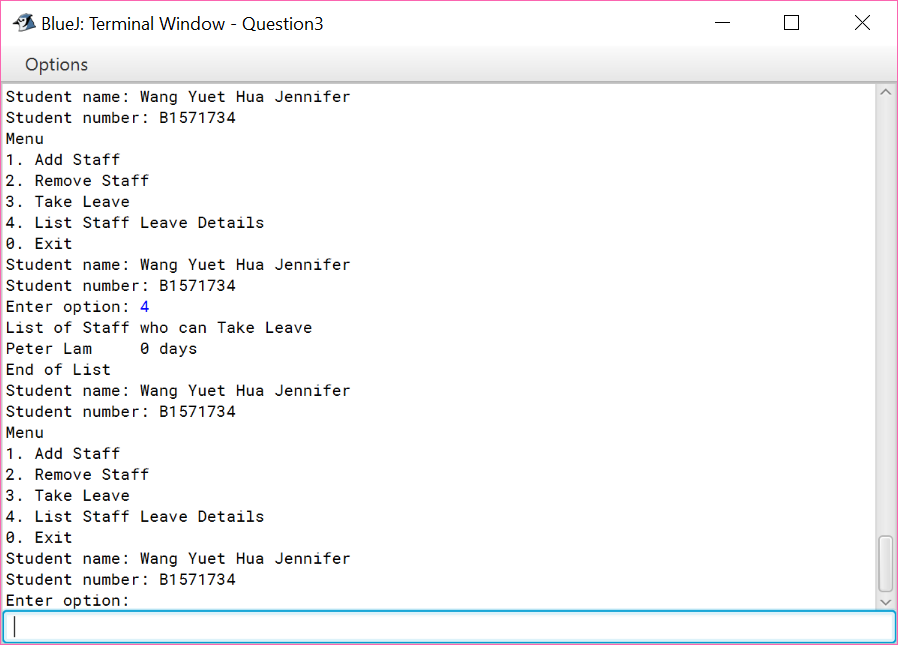
**Screenshot 8:**



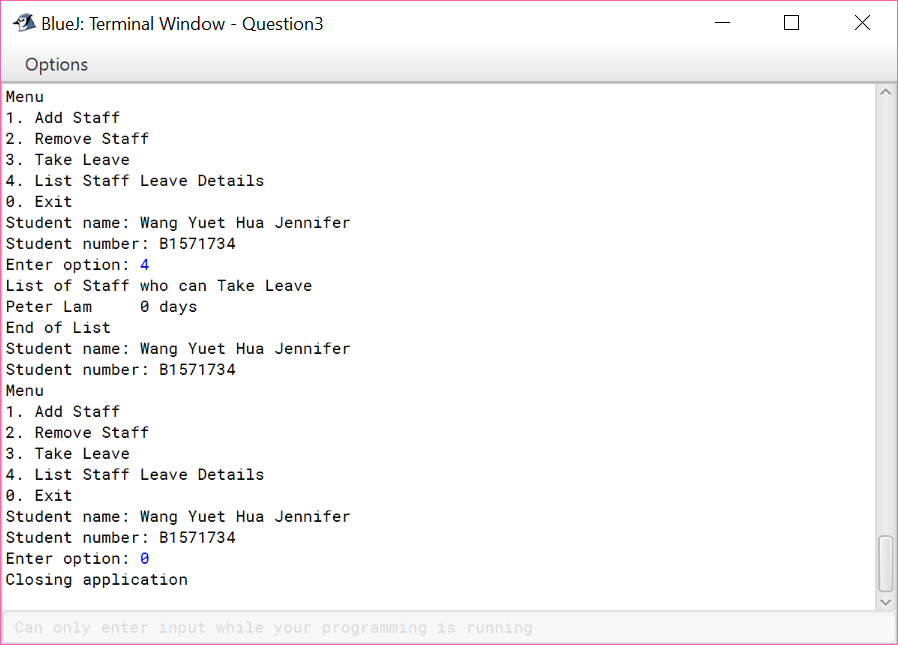
**Screenshot 9:**



**Screenshot 10:**



**Screenshot 11:**



**Question 4(a):**

class Staff

{

private String staffName;

private char staffCode;

private int balanceLeave;

private boolean canExchange;

//default constructor

public Staff(String name, int numLeave)

{

this(name, 'J', numLeave, false);

}

public Staff(String name, char code, int numLeave, boolean ableToExchange)

{

staffName = name;

staffCode = code;

balanceLeave = numLeave;

canExchange = ableToExchange;

}

public int getBalanceLeave() {

return balanceLeave;

}

public char getStaffCode() {

return staffCode;

}

public boolean getCanExchanged() {

return canExchange;

}

public void setBalanceLeave(int newBalanceLeave) {

balanceLeave = newBalanceLeave;

}

public void setStaffCode(char newStaffCode) {

staffCode = newStaffCode;

}

public void changeIsAllowedMoneyExchanged() {

canExchange = !canExchange;

}

private boolean hasLeave() {

if (balanceLeave>0)

return true;

else

return false;

}

private double computeLeaveToMoney(double rate) {

double amtMoney;

if (canExchange!= false) {

amtMoney = rate\*balanceLeave;

return amtMoney;

}

else

return -1;

}

private int compareTo(Staff firstStaff, int staffLeave) {

if (staffLeave>firstStaff.balanceLeave)

return 1;

else if (staffLeave<firstStaff.balanceLeave)

return -1;

else

return 0;

}

public String toString() {

double moneyConvertedFromLeave;

String description;

if (canExchange==true) {

moneyConvertedFromLeave = computeLeaveToMoney(112.50);

description = "The amount of money " + staffName + " converted from his " + balanceLeave + " day(s) of leave is $" + moneyConvertedFromLeave;

}

else

description = staffName + "is not allowed to exchange leave to money.";

return description;

}

}

**Question 4(b):**

import java.util.Scanner;

import java.util.ArrayList;

public class Question4

{

public static void main (String [] args)

{

String details;

ArrayList<Staff> staffList = new ArrayList<Staff>();

Staff firstStaff = new Staff("Dawn Ho", 14);

Staff secondStaff = new Staff("Alan Tan", 'S', 21, true);

staffList.add(firstStaff);

staffList.add(secondStaff);

staffList.get(0).changeIsAllowedMoneyExchanged();

staffList.get(1).setBalanceLeave(staffList.get(1).getBalanceLeave()-21);

for (int staff=0; staff<staffList.size(); staff++) {

if (staffList.get(staff).getBalanceLeave()>0)

System.out.println(staffList.get(staff).toString());

}

}

}

**Screenshot:**

