

Computer Assignment XII

(Second Selection)

18. An angle can be measured in degrees and minutes.

Example: Angle A = 70 degrees and 35 minutes

Angle B = 50 degrees and 40 minutes

Now find the sum of these two angles

Angle C = 70 degrees and 35 minutes + 50 degrees and 40 minutes
= 121 degrees and 14 minutes (As 1 degree = 60 minutes)

A class called Angle has been defined to calculate the angle related functions. Some of the functions/ methods are shown below:

Class Name: Angle

Data members/ Instance Variables:

- dd : Degrees
- mm : Minutes

Member functions

- Angle() : Constructor to initialize dd = 0, mm = 0;
- void readAngle() : To read angle as dd mm;
- void displayAngle() : To display angle as dd mm;
- void sumAngle(Angle A, Angle B) : To find sum of the two angles in degrees and minutes.
- void diffAngle(Angle A, Angle B) : To find the difference between the two angles in degrees and minutes.

Specify the class Angle giving details of the constructor, void sumAngle(Angle A, Angle B), void diffAngle(Angle A, Angle B) only. You may assume the other functions are written for you.

```
import java.util.*;
```

```
class Angle
```

```
{
```

```
    int dd, mm;
```

```
    Angle()
```

```
    {
```

```
        dd = 0;
```

```
        mm = 0;
```

```
    }
```

```
    void readAngle()
```

```
    {
```

```
        Scanner input = new Scanner(System.in);
```

```

System.out.print("Enter the angle in degree : ");
dd = input.nextInt();
System.out.print("Enter the angle in minutes : ");
mm = input.nextInt();
}
void displayAngle()
{
    System.out.println("Angle: " + dd + " degrees and " + mm + " minutes");
}
void sumAngle(Angle A, Angle B)
{
    this.dd = A.dd + B.dd;
    this.mm = A.mm + B.mm;
    if (this.mm >= 60)
    {
        this.mm = this.mm - 60;
        this.dd = this.dd + 1;
    }
}
void diffAngle(Angle A, Angle B)
{
    if (A.mm < B.mm)
    {
        A.dd = A.dd - 1;
        A.mm = A.mm + 60;
    }
    this.dd = A.dd - B.dd;
    this.mm = A.mm - B.mm;
}
public static void main(String Args[])
{
    Angle A = new Angle();
    A.readAngle();
    Angle B = new Angle();
    B.readAngle();

    Angle C = new Angle();

```

```

C.sumAngle(A,B);
System.out.println("The sum of the two angles is :");
C.displayAngle();

Angle D = new Angle();
D.diffAngle(A,B);
System.out.println("The difference between the two angles is: ");
D.displayAngle();
}
}

```

19. Specify the class WordPile giving the details of the constructor, void pushChar(char c) and char popChar(). The main function and the algorithm need not be written. Write a program in Java for conversion of WordPile into stack.

```

import java.util.*;
class WordPile
{
    int top, size;
    char ST[];
    WordPile(int n)
    {
        top = -1;
        size = n;
        ST = new char [size];
    }
    void pushChar(char c)
    {
        if (top == (size-1))
        {
            System.out.println("Overflow.");
        }
        else
        {
            top = top+1;
            ST [top] = c;
            System.out.println(c + " pushed into WordPile.");
        }
    }
}

```

```

    }
}
void popChar()
{
    if (top == -1)
    {
        System.out.println("The stack is empty.");
    }
    else
    {
        char v = ST [top];
        top = top-1;
        System.out.println(v + " character popped from WordPile");
    }
}
}
}

```

20. An array containing the marks of 50 students in ascending order is to be pushed into the stack. Define a class Array_to_Stack with the following details:

Class Name: Array_to_Stack

Data members/ Instance Variables

- int m : To store the marks
- int st[] : To store stack elements
- int cap : Maximum capacity of array and stack
- int top : To point the index of the topmost element of the stack

Member functions

- Array_to_Stack (int n) : Parametrized constructor to initialize cap = n and top = -1
- void inputMarks() : To input marks from the user and store it in the array n[] in ascending order and simultaneously push the marks into the stack st[] by invoking the function pushMarks()
- void pushMarks(int v) : To push the marks into the stack at top location if possible otherwise display "Not Possible"
- int popMarks() : To return marks from the stack if possible otherwise return 9999
- void display() : To display stack element

Specify the class Array_to_Stack giving details of the constructor(int n), void inputMarks(), void pushMarks(), void popMarks() and void display().

```

import java.util.*;

class Array_to_Stack

```

```

{
    int m;
    int n[];
    int st[];
    int cap;
    int top;
    Array_to_Stack(int n)
    {
        cap = n;
        top = -1;
    }
    void inputMarks()
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the marks of the students.");
        for (int i = 0; i < 50; i++)
        {
            System.out.print("Marks for " + (i+1) + " : ");
            n[i] = input.nextInt();
            pushMarks(n[i]);
        }
        Arrays.sort(st);
    }
    void pushMarks( int v)
    {
        if (top == (cap-1))
        {
            System.out.println("Not Possible.");
        }
        else
        {
            top = top+1;
            st[top] = v;
        }
    }
    void popMarks()
    {

```

```

if (top == -1)
{
    System.out.println("9999");
}
else
{
    int b = st[top];
    top = top-1;
    System.out.println(b+" popped out of the stack.");
}
}

void display()
{
    System.out.println("The marks scored by 50 students are : ");
    for(int i=0; i<=top; i++)
    {
        System.out.print((i+1)+ " : "+st[i]+"\\n");
    }
}

public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    Array_to_Stack object = new Array_to_Stack(50);
    object.inputMarks();
    while(true)
    {
        System.out.println("\\nMenu for Array to Stack");
        System.out.println("1.Pop items");
        System.out.println("2.Display Stack");
        System.out.println("3.Exit");
        System.out.print("Enter your choice: ");
        int choice = input.nextInt();
        switch(choice)
        {
            case 1:
                object.popMarks();
                break;

```

```

        case 2:
            object.display();
            break;
        case 3:
            System.out.println("Exiting the program.");
            System.exit(0);
            break;
        default:
            System.out.println("Invalid choice.");
    }
}
}
}

```

21. Write a program in java to make various operations in a queue.

```

import java.util.*;

class Queue_Operations
{
    int rear;
    int front;
    int size;
    int q[];

    Queue_Operations(int n)
    {
        rear = -1;
        front = -1;
        q = new int[n];
        size = n;
    }

    void enqueue(int item)
    {
        if (rear == (size-1))
        {
            System.out.println("Queue Overflow.");
        }
        else

```

```

{
    if(front == -1)
    {
        front = 0;
    }
    rear = rear+1;
    q[rear] = item;
}
}

void dequeue()
{
    if(front == -1)
    {
        System.out.println("The queue is empty.");
    }
    else
    {
        if(front>rear)
        {
            front = -1;
            rear = -1;
        }
        int dq = q[front];
        front++;
        System.out.println(dq+" dequeued from the queue.");
    }
}

void display()
{
    if (front == -1)
    {
        System.out.println("The queue is empty.");
    }
    else
    {
        System.out.println("Elements of the queue are : ");
        for (int i=front ; i<=rear ; i++ )

```



```

        {
            System.out.print(q[i]+" ");
        }
        System.out.println();
    }
}

public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the size of the array : ");
    int s = input.nextInt();
    Queue_Operations object = new Queue_Operations(s);
    while (true)
    {
        System.out.println("\nQueue Operations:");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Display Queue");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        int choice = input.nextInt();
        switch (choice)
        {
            case 1:
                System.out.print("Enter the element to enqueue: ");
                int element = input.nextInt();
                object.enqueue(element);
                break;
            case 2:
                object.dequeue();
                break;
            case 3:
                object.display();
                break;
            case 4:
                System.out.println("Exiting the program.");
                System.exit(0);
        }
    }
}

```

```

        break;
    default:
        System.out.println("Invalid choice.");
    }
}
}
}
}

```

22. A class Student defines student related information such as name, roll number and date of birth while another class Marks defines marks in various subjects, total percentage and grade of the student.

The details of the classes are given below:

Class name : Student

Data members/ Instance Variables

- String name, DOB : String variables to store name and date of birth of student.
- int roll_num : Integer variable to store roll number

Member functions

- void inputData() : To input values of all the data members.
- void printData() : To display the values of all data members.

Class name : Marks

Data members/ Instance Variables

- int phy, chem, maths, cs, eng, total, per
- double per

Member functions

- void readData() : To read marks in all subjects out of 100
- void compute() : To find and store total marks, percentage and grade
- void printData() : To display the details

Specify the class Student with its functions. Using concept of inheritance specify the class marks with its functions.

Class Marks is derived from class Student.

```

import java.util.*;
class Student
{

```

```

String name, DOB;
int roll_num;
void inputData()
{
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the name of the student : ");
    name = input.next();
    System.out.print("Enter the date of birth of the student : ");
    DOB = input.next();
    System.out.print("Enter the roll number of the student");
    roll_num = input.nextInt();
}
void printData()
{
    System.out.println("Name : "+name);
    System.out.println("Date of Birth : "+DOB);
    System.out.println("Roll no. : "+roll_num);
}
}

import java.util.*;
class Marks extends Student
{
    int phy, chem, maths, cs, eng, total;
    double per;

    void readData()
    {
        super.inputData();
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the marks of the given subjects : ");
        System.out.print("Physics : ");
        phy = input.nextInt();
        System.out.print("Chemistry : ");
        chem = input.nextInt();
        System.out.print("Mathematics : ");
        maths = input.nextInt();
    }
}

```

```

    System.out.print("Computer Science : ");
    cs = input.nextInt();
    System.out.print("English : ");
    eng = input.nextInt();
}
void compute()
{
    total = (phy + chem + maths + cs + eng);
    per = (total/500.0)*100;
}
void printData()
{
    super.printData();
    System.out.print("Physics : " +phy);
    System.out.print("Chemistry : " +chem);
    System.out.print("Mathematics : " +maths);
    System.out.print("Computer Science : " +cs);
    System.out.print("English : " +eng);
    System.out.print("Total : " +total);
    System.out.print("Percentage : " +per);
}
public static void main(String Args[])
{
    Marks object = new Marks();
    object.readData();
    object.compute();
    object.printData();
}
}

```

23. A super class Detail has been defined to store the details of the customer.

Define a subclass Bill to compute the monthly telephone charge of the customer as per the chart given below:

Number of calls	Rate
1 - 100	Only rental charge
101- 200	60 paise per call + rental charge
200 - 300	80 paise per call + rental charge

Above 300

1 rupee per call + rental charge

The details of both classes are given below

Class name : Detail

Data members/ Instance Variables

- String name : To store the name of the customer
- String address : To store the address of the customer
- long tel_no : To store the telephone number of the customer
- double rental : To store the monthly rental charge

Member functions

- Detail() : Parametrized constructor to assign values to the data members
- void show() : To show the details of the customer

Class name : Bill

Data members/ Instance Variables

- int n : To store the number of phone calls
- double amt : To store the amount to be paid by the customer

Member functions

- Bill() : Parametrized constructor to assign values to data members and to initialize amt = 0.0
- void call() : Calculates the monthly telephone bill as per the charge given above
- void show() : To display the detail of the customer and amount to be paid.

Specify the class Detail giving details of the constructor() and void show().

Using the concept of inheritance , specify the class Bill giving details of the constructor(), void call() and void show().

The main function and algorithm need not be written.

```
import java.util.*;

class Detail
{
    String name, address;
    long tel_no;
    double rental;
    Detail(String n, String a, long t, double r)
    {
```

```

    name = n;
    address = a;
    tel_no = t;
    rental = r;
}
void show()
{
    System.out.println("Name : "+name);
    System.out.println("Address : "+address);
    System.out.println("Telephone number : "+tel_no);
    System.out.println("Rental Charge : "+rental);
}
}

```

```

import java.util.*;
class Bill extends Detail
{
    int n;
    double amt;
    Bill(String nam, String add, long tel, double ren, int call)
    {
        super(nam,add,tel,ren);
        n = call;
        amt = 0.0;
    }
    void call()
    {
        if(n>=1 && n<=100)
        {
            amt = rental;
        }
        else if (n>=101 && n<=200)
        {
            amt = (0.6*n)+rental;
        }
        else if (n>=200 && n<=300)
        {

```

```

        amt = (0.8*n)+rental;
    }
    else
    {
        amt = n + rental;
    }
}
void show()
{
    super.show();
    System.out.println("The amount to be paid by the customer :"+amt);
}
}

```

24. A class Admission contains the admission number of 100 students. Some of the data members/ member functions are given below:

Class name: Admission

Data members/ Instance Variables

- int Ad_no[] : Integer array to store admission number

Member functions

- Admission() : Constructor to initialize the array elements
- void fill_Array : To accept element of array and rearrange in ascending order.
- int binSearch(int l, int u, int v) : To search for particular admission number(v) using binary search and recursive techniques and return 1 if found and -1 if not found

Specify the class Admission giving details of the constructor, void fill_Array() and int binSearch(int l, int u, int v).

Define the main function and call the functions according to enable the task.

```

import java.util.*;
class Admission
{
    int Ad_no[];
    Admission()
    {
        Ad_no = new int[100];
    }
}

```

```

void fill_array()
{
    Scanner input = new Scanner(System.in);
    System.out.println("Enter the admission number of the students : ");
    for(int i=0; i<100; i++)
    {
        System.out.print "["+(i+1)+"] : ";
        Ad_no[i] = input.nextInt();
    }
    Arrays.sort(Ad_no);
}

int binSearch(int l, int u, int v)
{
    if (l <= u)
    {
        int mid = l + (u - l) / 2;
        if (Ad_no[mid] == v)
        {
            return 1;
        }
        else if (Ad_no[mid] < v)
        {
            return binSearch(mid + 1, u, v);
        }
        else
        {
            return binSearch(l, mid - 1, v);
        }
    }
    return -1;
}

public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    Admission object = new Admission();
    object.fill_array();
    System.out.print("Enter the admission number you want to search for : ");
}

```



```
int search = input.nextInt();
int result = object.binSearch(0,99,search);
if(result == -1)
{
    System.out.println(search+" is not present in the list of admission number");
}
else
{
    System.out.println(search+" is present in the list of admission number");
}
}
```