EX.NO:1 Solve problems by usings sequential search,binary searchand quadratic algorithms (selectio,insertion)

AIM:- To write java program to perfore the insertion short

ALGORITHM:

1.Start

2.Iterate from arr[1] to arr[N] over the array.

3.Compare the current element (key) to its predecessor.

4.If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.

5.Stop

PROGRAM:-

```
import java.util.Arrays;
class InsertionSort {
  void insertionSort(int array[]) {
    int size = array.length;
    for (int step = 1; step < size; step++) {
      int key = array[step];
      int j = step - 1;
     while (j >= 0 && key < array[j]) {
        array[j + 1] = array[j];
        --j;
      }
    array[j + 1] = key;
    }
  }
 public static void main(String args[]) {
    int[] data = { 9, 5, 1, 4, 3 };
    InsertionSort is = new InsertionSort();
```

```
    is.insertionSort(data);
    System.out.println("Sorted Array in
Ascending Order: ");
    System.out.println(Arrays.toString(data));
  }
}
```

OUT PUT:-

Sorted Array in Ascending Order:

[1, 3, 4, 5, 9]

RESULT:-

The above java is executed successfully and output is verified

---

//selection short

AIM:- To write the java program to performe the selection short.

Algorithm:

Step 1 – Start

Step 2 – Set Min_Index to 0

Step 3 − Search for the smallest element in the array

Step 4 − Swap with value with the element at the Min_Index

Step 5 − Increment Min_Index to point to next element

Step 6 − Repeat until the complete array is sorted

Step 7 − Stop

PROGRAM:-

```
import java.util.Arrays;
class selectioshort {
        void selectionSort(int array[]) {
          int size = array.length;

          for (int step = 0; step < size - 1; step++) {
            int min_idx = step;
```

```
        for (int i = step + 1; i < size; i++) {
          if (array[i] < array[min_idx]) {
            min_idx = i;
          }
        }
        int temp = array[step];
        array[step] = array[min_idx];
        array[min_idx] = temp;
      }
    }
    public static void main(String args[]) {
      int[] data = { 20, 12, 10, 15, 2 };
      king ss = new king();
      ss.selectionSort(data);
      System.out.println("Sorted Array in
Ascending Order: ");
```

```
System.out.println(Arrays.toString(data));
    }
}
```

OUTPUT:-

Sorted Array in Ascending Order:

[2, 10, 12, 15, 20]

RESULT:-

The above java is executed successfully and output is verified

---

EX.NO:2 Develop stack and queue data structures using classes and object

// Stack implementation in Java

AIM:- TO Develop the java program to implement the stack.

Algorithm:

Step 1 – Start

Step 2 − Checks if the stack is full.

Step 3 − If the stack is full, produces an error and exit.

Step 4 − If the stack is not full, increments top to point next empty space.

Step 5 − Adds data element to the stack location, where top is pointing.

Step 6 − Returns success.

Step 7 – Stop

```
class Stack {
  private int arr[];
  private int top;
  private int capacity;
  Stack(int size) {
    arr = new int[size];
    capacity = size;
    top = -1;
  }
public void push(int x) {
    if (isFull()) {
```

```
      System.out.println("OverFlow\nProgram
Terminated\n");
      System.exit(1);
    }
    System.out.println("Inserting " + x);
    arr[++top] = x;
  }
  public int pop() {
    if (isEmpty()) {
      System.out.println("STACK EMPTY");
      System.exit(1);
    }
    return arr[top--];
  }
public int size() {
    return top + 1;
  }
  public Boolean isEmpty() {
    return top == -1;
  }
```

```
  public Boolean isFull() {
    return top == capacity - 1;
  }
  public void printStack() {
    for (int i = 0; i <= top; i++) {
      System.out.println(arr[i]);
    }
  }
public static void main(String[] args)
{
   Stack stack = new Stack(5);
    stack.push(1);
    stack.push(2);
    stack.push(3);
    stack.push(4);
    stack.pop();
    System.out.println("\nAfter popping out");
  stack.printStack();
  }
}
```

Output:

Inserting 1

Inserting 2

Inserting 3

Inserting 4

After popping out

1

2

3

RESULT:-

The above java programe is executed successfully and output is verified

---

AIM:-TO develop the java program for implementation of queue

Algorithm:

Step 1 – Start

Step 2 − Check if the queue is full.

Step 3 − If the queue is full, produce overflow error and exit.

Step 4 − If the queue is not full, increment rear pointer to point the next empty space.

Step 5 − Add data element to the queue location, where the rear is pointing.

Step 6 − return success.

Step 7 − Stop

PROGRAME:-

```
// Queue implementation in Java
public class Queue {
  int SIZE = 5;
  int items[] = new int[SIZE];
  int front, rear;
 Queue() {
    front = -1;
    rear = -1;
  }
 Boolean isFull() {
    if (front == 0 && rear == SIZE - 1) {
```

```
    return true;
  }
  return false;
}
Boolean isEmpty() {
  if (front == -1)
    return true;
  else
    return false;
}
void enQueue(int element) {
  if (isFull()) {
    System.out.println("Queue is full");
  } else {
    if (front == -1)
      front = 0;
    rear++;
    items[rear] = element;
    System.out.println("Inserted " + element);
  }
```

```
  }
  int deQueue() {
    int element;
    if (isEmpty()) {
      System.out.println("Queue is empty");
     return (-1);
    } else {
      element = items[front];
      if (front >= rear) {
        front = -1;
        rear = -1;
      }
  else {
        front++;
      }
      System.out.println("Deleted -> " + element);
      return (element);
    }
  }
void display() {
```

```
    int i;
    if (isEmpty()) {
      System.out.println("Empty Queue");
    }
  else {
      System.out.println("\nFront index-> " + front) ;
System.out.println("Items -> ");
      for (i = front; i <= rear; i++)
        System.out.print(items[i] + "  ");

      System.out.println("\nRear index-> " + rear);
    }
  }
 public static void main(String[] args) {
    Queue q = new Queue();
   q.deQueue();
   q.enQueue(1);
    q.enQueue(2);
    q.enQueue(3);
    q.enQueue(4);
```

```
    q.enQueue(5);
    q.enQueue(6);
    q.display();
     q.deQueue();
    q.display();
   }
  }
```

OUTPUT:-

```
Queue is empty
Inserted 1
Inserted 2
Inserted 3
Inserted 4
Inserted 5
Queue is full
Front index-> 0
Items ->
1  2  3  4  5
Rear index-> 4
Deleted -> 1Front index-> 1
```

Items ->

2 3 4 5

Rear index-> 4

RESULT:-

The above java program is executed successfully and output is verified

---

EX.NO:3 Develop a java application with an Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary.

AIM:-To develop the java programe for employee using inheritences.

ALGORITHM:-

Step1-Start

Step2- Create a employee class in which declare the variable.

Step3- Extend the employee class to the programmer,

Assistant professor,associate professor and professor

Classes.

Step4- In each class give the print statement.

Step5- In main class use switch statement and create the object to respective classes.

Step6- Using object assign the value to variables.

Step7- Display the result.

Step8-Stop.

PROGRAME:-

```
Import java.io.*;
import java.util.Scanner;
 class Employee {

    String Emp_name;
        int Emp_id;
        String Address;
        String Mail_id;
        long Mobile_no;
}
class programmer extends Employee{
        int BP;
```

```
        double net_amount;
        void display() {
            System.out.println("NAME="+Emp_name);
            System.out.println("ID="+Emp_id);
            System.out.println("ADDRESS="+Address);
            System.out.println("G-MAIL="+Mail_id);
            System.out.println("PHONE NO="+Mobile_no);
                System.out.println("BASIC_PAY="+BP);
                System.out.println("DA="+0.97*BP);
                System.out.println("HRA="+0.10*BP);
                System.out.println("PF="+0.12*BP);
                System.out.println("staff club
funds="+0.001*BP);


net_amount=(0.97*BP)+(0.10*BP)+(0.12*BP)+(0.001*B
P);

        System.out.println("Net_salary="+net_amount);


        }
```

```
}
class Assistant_Professor extends Employee{
        int BP;
        double net_amount;
        void display() {
        System.out.println("NAME="+Emp_name);
           System.out.println("ID="+Emp_id);
           System.out.println("ADDRESS="+Address);
           System.out.println("G-MAIL="+Mail_id);
           System.out.println("PHONE NO="+Mobile_no);
              System.out.println("BASIC_PAY="+BP);
              System.out.println("DA"+0.97*BP);
              System.out.println("HRA"+0.10*BP);
              System.out.println("PF"+0.12*BP);
              System.out.println("staff club
funds"+0.001*BP);


net_amount=(0.97*BP)+(0.10*BP)+(0.12*BP)+(0.001*B
P);
```

```
        System.out.println("Net_salary="+net_amount);
        }
        }
class Associate_Professor extends Employee{
        int BP;
        double net_amount;
        void display() {
        System.out.println("NAME="+Emp_name);
           System.out.println("ID="+Emp_id);
           System.out.println("ADDRESS="+Address);
           System.out.println("G-MAIL="+Mail_id);
           System.out.println("PHONE NO="+Mobile_no);
              System.out.println("BASIC_PAY="+BP);
              System.out.println("DA"+0.97*BP);
              System.out.println("HRA"+0.10*BP);
              System.out.println("PF"+0.12*BP);
              System.out.println("staff club
funds"+0.001*BP);
```

```
net_amount=(0.97*BP)+(0.10*BP)+(0.12*BP)+(0.001*B
P);

        System.out.println("Net_salary="+net_amount);
        }
        }
class professor extends Employee{
        int BP;
        double net_amount;
        void display() {
        System.out.println("NAME="+Emp_name);
           System.out.println("ID="+Emp_id);
           System.out.println("ADDRESS="+Address);
           System.out.println("G-MAIL="+Mail_id);
           System.out.println("PHONE NO="+Mobile_no);
              System.out.println("BASIC_PAY="+BP);
              System.out.println("DA"+0.97*BP);
              System.out.println("HRA"+0.10*BP);
              System.out.println("PF"+0.12*BP);
              System.out.println("staff club
funds"+0.001*BP);
```

```
net_amount=(0.97*BP)+(0.10*BP)+(0.12*BP)+(0.001*BP);

        System.out.println("Net_salary="+net_amount);
        }
        }

class Sample{
        public static void main(String[] args) {


System.out.println("1.programmer\n2.Assistant_Professor\n3.Associative_Professor\n4.professor");
        Scanner o=new Scanner(System.in);
        int no;
        System.out.println("\nenter the person no to print the details");
        no=o.nextInt();
        switch(no) {
        case 1:
```

```
        programmer e1=new programmer();
        e1.Emp_name="kumar";
        e1.Address="chennai";
        e1.Mail_id="kumar123@";
        e1.Emp_id=123;
        e1.Mobile_no=1;
        e1.BP=10000;
        e1.display();
        break;
        case 2:
Assistant_Professor e2=new Assistant_Professor();
        e2.Emp_name="raja";
        e2.Address="madurai";
        e2.Mail_id="raja124@";
        e2.Emp_id=124;
        e2.Mobile_no=2;
        e2.BP=20000;
        e2.display();
         break;
        case 3:
```

```
Associate_Professor e3=new Associate_Professor();
        e3.Emp_name="santhosh";
        e3.Address="trichy";
        e3.Mail_id="santhosh125@";
        e3.Emp_id=125;
        e3.Mobile_no=3;
        e3.BP=30000;
        e3.display();
        break;
        case 4:
 professor e4=new  professor();
        e4.Emp_name="siva";
        e4.Address="dindigul";
        e4.Mail_id="siva125@";
        e4.Emp_id=125;
        e4.Mobile_no=4;
        e4.BP=40000;
        e4.display();
        break;
        default:System.out.println("enter the correct no");
```

```
    }
    }}
```

OUTPUT:-

```
1.programmer
2.Assistant_Professor
3.Associative_Professor
4.professor
enter the person no to print the details
1
NAME=kumar
ID=123
ADDRESS=chennai
G-MAIL=kumar123@
PHONE NO=1
BASIC_PAY=10000
DA=9700.0
HRA=1000.0
PF=1200.0
staff club funds=10.0
```

Net_salary=11910.0

RESULT:-

The above java programe is executed successfully and output is verified

EX.NO:4:- Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape.

AIM:-To Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape.

ALGORITHM:-

Step1- Start.

Step2- create a abstract class shape with apstract method.
Step3- extend the shape to triangle , rectangle and circle.
Step4- call the method using creating the object to the classes.
Step5- display the area of the shapes.
Step6- stop.

PROGRAM:-

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void area(double x,double y);
}
class Rectangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
{
```

```
void area(double x,double y)
{
System.out.println("Area of triangle is :"+(0.5*x*y));
}
}
public class  program
{
public static void main(String[] args)
{
Rectangle r=new Rectangle();
r.area(2,5);
Circle c=new Circle();
c.area(5,5);
Triangle t=new Triangle();
t.area(2,5);
}
}
```

Output:-

Area of rectangle is :10.0

Area of circle is :78.5

Area of triangle is :5.0

RESULT:-

The above java programe is executed successfully and output is verified

EX NO 5:- Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape using interface.

AIM:- TO Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shapeUSING INTERFACE.

ALGORITHM:-

Step1- start.

Step2- create a interface shape and initialize the value as zero and declare the method.

Step3- extend the shape to triangle , rectangle and circle.

Step4- call the method using creating the object to the classes.

Step5- display the area of the shapes.

Step6- stop.

PROGRAM:-

```
interface  shape
{
int x=0,y=0;
 void area(double x,double y);
}
class Rectangle implements shape
{
public void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle implements shape
{
public void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
```

```
}
}
class Triangle implements shape
{

	public void area(double x,double y)
	{
	System.out.println("Area of triangle is
:"+(0.5*x*y));
	}
	}
	public class  sample
	{
	public static void main(String[] args)
	{
	Rectangle r=new Rectangle();
	r.area(2,5);
	Circle c=new Circle();
	c.area(5,5);
	Triangle t=new Triangle();
```

```
        t.area(2,5);
        }
}
```

OUTPUT:-

Area of rectangle is :10.0

Area of circle is :78.5

Area of triangle is :5.0

RESULT:-

The above java programe is executed successfully and output is verified

EX NO 6:- Java program to implement user defined exception handling

AIM:- To write a Java program for implementing user defined exception handling.

ALGORITHM:-

Step1- Start.

Step2- create a class handling.

Step3- Assign the value to a,b,c.

Step4- Give the error int the Try block.

Step5- give the errror statement in the catch block.

Step6- stop.

PROGRAM:-

```
class handling
{
	public static void main(String args[])
	{
		int a=10,b=0;
		int c=0;
		try
		{
			c=a/b;
		}
		catch(Exception e)
		{
```

```
                System.out.println("we can not able to
run sorry!!!");


            }
            System.out.println("c="+c);
            System.out.println("the program is
executed");
      }
}
```

OUTPUT:-

```
we can not able to run sorry!!!
c=0
the program is executed
```

RESULT:-

The above java programe is executed successfully and output is verified

EXNO 7:- write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the

square of the number and prints. If the value is odd, the third thread will print the value of cube of the number

AIM:- To write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

PROGRAME:-

```
import java.util.*;
// class for Even Number
class EvenNum implements Runnable {
   public int a;
   public EvenNum(int a) {
      this.a = a;
   }
   public void run() {
      System.out.println("The Thread "+ a +" is EVEN and Square of " + a + " is : " + a * a);
   }
```

```
} // class for Odd Number
class OddNum implements Runnable {
   public int a;
   public OddNum(int a)  {
      this.a = a;
   }
   public void run()  {
      System.out.println("The Thread "+ a +" is ODD and
Cube of " + a + " is: " + a * a * a);
   }
}
// class to generate random number
class RandomNumGenerator extends Thread  {
   public void run() {
      int n = 0;
      Random rand = new Random();
      try  {
         for (int i = 0; i < 10; i++) {
            n = rand.nextInt(20);
            System.out.println("Generated Number is " +
n);
```

```
            // check if random number is even or odd
            if (n % 2 == 0) {
                Thread thread1 = new Thread(new
EvenNum(n));
                thread1.start();
            }
            else {
                Thread thread2 = new Thread(new
OddNum(n));
                thread2.start();
            }
        // thread wait for 1 second
        Thread.sleep(1000);
        System.out.println("---------------------------------");
            }
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

```
// Driver class
public class keerthi {
    public static void main(String[] args) {
        RandomNumGenerator rand_num = new
RandomNumGenerator();
        rand_num.start();
}
}
```

Output

```
Generated Number is 10
The Thread 10 is EVEN and Square of 10 is : 100
----------------------------------
Generated Number is 17
The Thread 17 is ODD and Cube of 17 is: 4913
----------------------------------
Generated Number is 11
The Thread 11 is ODD and Cube of 11 is: 1331
----------------------------------
Generated Number is 15
The Thread 15 is ODD and Cube of 15 is: 3375
```

Generated Number is 12

The Thread 12 is EVEN and Square of 12 is : 144

Generated Number is 18

The Thread 18 is EVEN and Square of 18 is : 324

Generated Number is 11

The Thread 11 is ODD and Cube of 11 is: 1331

Generated Number is 18

The Thread 18 is EVEN and Square of 18 is : 324

Generated Number is 12

The Thread 12 is EVEN and Square of 12 is : 144

Generated Number is 1

The Thread 1 is ODD and Cube of 1 is: 1

RESULT:-

The above java programe is executed successfully and output is verified

EXNO 8:-Write a program to performe file operations

AIM:-

TO Write a program to performe file operations.

ALGORITHM:-

Step1- Start.

Step2- create a object to a create a new file.

Step3- wrie the statement in the same file.

Step4- print the file path.

Step5- print the file name,size and content.

Step6- Stop.

PROGRAME:-

```
import java.io.*;
import java.io.IOException;
```

```
import java.util.Scanner;


class sample{
   public static void main(String args[]) throws
IOException
   {
  File f0 = new File("D:FileOperationExample2.txt");
if (f0.createNewFile()) {
 System.out.println("File " + f0.getName() + " is created
successfully.");
   } else {
          System.out.println("File is already exist in the
directory.");
  }
FileWriter fw=new
FileWriter("D:FileOperationExample2.txt");
fw.write("hi welcome to java programe");
fw.close();
System.out.print("name="+f0.getName());


System.out.println("path="+f0.getAbsolutePath());
```

```
System.out.println("size of file is:="+f0.length());
String filedata;
Scanner dataReader=new Scanner(f0);
filedata=dataReader.nextLine();
System.out.println(filedata);
}
}
```

OUTPUT:-

```
File FileOperationExample2.txt is created successfully.
name=FileOperationExample2.txt
path=D:\\FileOperationExample2.txt
size of file is:=27
hi welcome to java programe
```

RESULT:-

The above java programe is executed successfully and output is verified

EXNO 9:-Develop the application to demonstrate the features of generics classes.

AIM:-TO Develop the application to demonstrate the features of generics classes.

ALGORITHM:-

PROGRAM:-

```
public class area<T>
{
	private T t;
	public void add(T t)
	{
		this.t=t;
	}
	public T get(){return t;}
	public void getarea() {}
	public static void main(String args[])
	{
		area<Integer>rectangle=new area<Integer>();
		area<Double>circle= new area<Double>();
```

```
        rectangle.add(10);
        circle.add(2.5);
        System.out.println(rectangle.get());
        System.out.println(circle.get());
    }

}
```

OUTPUT:-

10

2.5

RESULT:-

The above java programe is executed successfully and output is verified