

# **Object Oriented Programming Lab**

## **(CS 32203)**

### **Assignment - 4**

**Name: Sandeep Kumar Singh**

**Reg. No: 2021CA094**

**1. Write a java program to implement quick sort.**

```
import java.util.Scanner;

public class QuickSort {
    static void swap(int[] arr, int i, int j)
    {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    static int partition(int[] arr, int low, int high)
    {
        int pivot = arr[high];
        int i = (low - 1);
```

```
        for(int j = low; j <= high - 1; j++)
        {

            if (arr[j] < pivot)
            {
                i++;
                swap(arr, i, j);
            }
        }
        swap(arr, i + 1, high);
        return (i + 1);
    }

    static void quickSort(int[] arr, int low, int high)
    {
        if (low < high)
        {
            int pi = partition(arr, low, high);

            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }

    static void printArray(int[] arr)
```

```
{
    for (int value : arr)
        System.out.print(value + " ");
    System.out.println();
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter size of array : ");
    int size = sc.nextInt();
    int[] arr = new int[size];
    System.out.println("Enter array elements : ");
    for (int i=0; i<size; i++)
        arr[i] = sc.nextInt();

    quickSort(arr, 0, size - 1);
    System.out.println("Sorted array: ");
    printArray(arr);
}
}
```

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
Enter size of array : 10
Enter array elements :
423 543 32 65 324 34 23 4 24 67
Sorted array:
4 23 24 32 34 65 67 324 423 543
Process finished with exit code 0
```

**2. Write a java program to implement Merge sort.**

```
import java.util.Scanner;

public class MergeSort
{
    void merge(int[] arr, int l, int m, int r)
    {
        int n1 = m - l + 1;
        int n2 = r - m;

        int[] L = new int[n1];
        int[] R = new int[n2];

        for (int i = 0; i < n1; ++i)
            L[i] = arr[l + i];
        for (int j = 0; j < n2; ++j)
            R[j] = arr[m + 1 + j];

        int i = 0, j = 0;

        int k = l;
        while (i < n1 && j < n2) {
            if (L[i] <= R[j]) {
                arr[k] = L[i];
```

```
        i++;  
    }  
    else {  
        arr[k] = R[j];  
        j++;  
    }  
    k++;  
}  
  
while (i < n1) {  
    arr[k] = L[i];  
    i++;  
    k++;  
}  
while (j < n2) {  
    arr[k] = R[j];  
    j++;  
    k++;  
}  
}  
  
void sort(int[] arr, int l, int r)  
{  
    if (l < r) {
```

```
        int m = l + (r - l) / 2;

        sort(arr, l, m);
        sort(arr, m + 1, r);

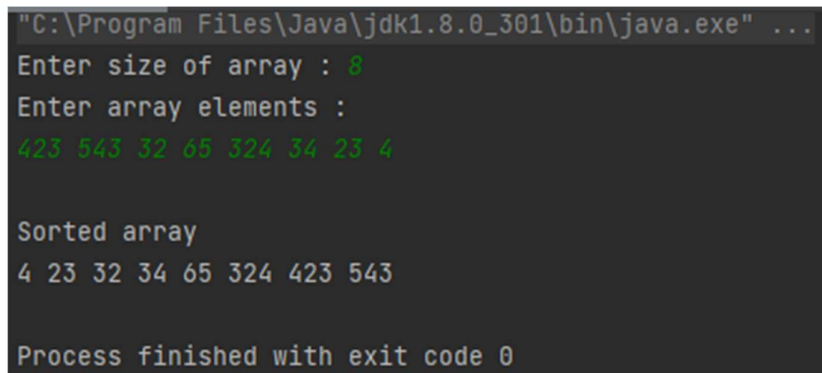
        merge(arr, l, m, r);
    }
}

static void printArray(int[] arr)
{
    for (int value : arr)
        System.out.print(value + " ");

    System.out.println();
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter size of array : ");
    int size = sc.nextInt();
    int[] arr = new int[size];
    System.out.println("Enter array elements : ");
```

```
        for (int i=0; i<size; i++)  
            arr[i] = sc.nextInt();  
  
        MergeSort ob = new MergeSort();  
        ob.sort(arr, 0, size - 1);  
  
        System.out.println("\nSorted array");  
        printArray(arr);  
    }  
}
```



A screenshot of a Java program execution in a command prompt. The prompt shows the path "C:\Program Files\Java\jdk1.8.0\_301\bin\java.exe" followed by the program's output. The user is prompted to enter the size of the array (8) and the array elements (423 543 32 65 324 34 23 4). The program then displays the sorted array (4 23 32 34 65 324 423 543) and finishes with exit code 0.

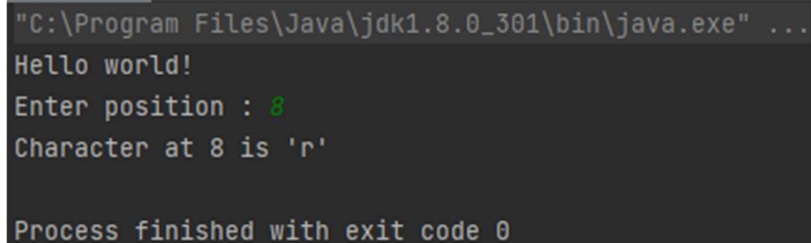
```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...  
Enter size of array : 8  
Enter array elements :  
423 543 32 65 324 34 23 4  
  
Sorted array  
4 23 32 34 65 324 423 543  
  
Process finished with exit code 0
```

**3. Write a Java program to get the character at the given index within the String**

```
import java.util.Scanner;  
  
public class FindChar {  
    public static char findChar(String s,int position){  
        return s.charAt(position);  
    }  
}
```

```
}
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    String str = "Hello world!";  
    System.out.println(str);  
    System.out.print("Enter position : ");  
    int position = sc.nextInt();  
  
    System.out.println("Character at " +position+ " is  
    "+findChar(str,position)+"");  
}  
}
```



A screenshot of a Windows command prompt window showing the execution of a Java program. The command prompt title bar reads "C:\Program Files\Java\jdk1.8.0\_301\bin\java.exe" ... The output of the program is as follows: "Hello world!" on the first line, "Enter position : 8" on the second line (where '8' is entered in green), "Character at 8 is 'r'" on the third line, and "Process finished with exit code 0" on the fourth line.

**4. Write a Java program to trim any leading or trailing whitespace from a given string**

```
import java.util.Scanner;  
  
public class RemoveWhiteSpace {  
    public static void main(String[] args) {
```



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

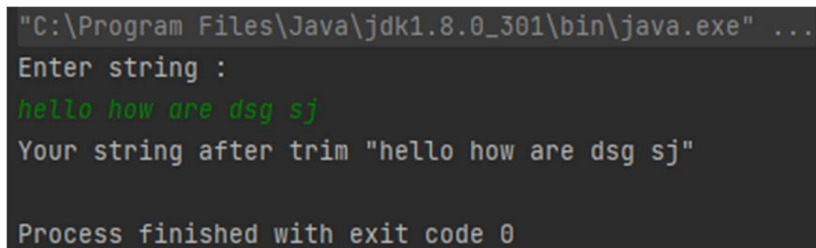
System.out.println("Enter string : ");

String name = sc.nextLine();

System.out.println("Your string after trim
\""+name.trim()+"\");

}

}
```

A screenshot of a terminal window showing the execution of a Java program. The window title is "C:\Program Files\Java\jdk1.8.0\_301\bin\java.exe" ... The output shows the prompt "Enter string :", the user input "hello how are dsg sj" in green, and the program output "Your string after trim \"hello how are dsg sj\"". At the bottom, it says "Process finished with exit code 0".

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
Enter string :
hello how are dsg sj
Your string after trim "hello how are dsg sj"
Process finished with exit code 0
```

5. Write a Java program to compare two strings lexicographically. Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions

```
import java.util.Scanner;

public class CompareString {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter fist string : ");

        String str1 = sc.nextLine();

        System.out.println("Enter second string : ");

        String str2 = sc.nextLine();
```

```
        if (str1.compareTo(str2) == 0)

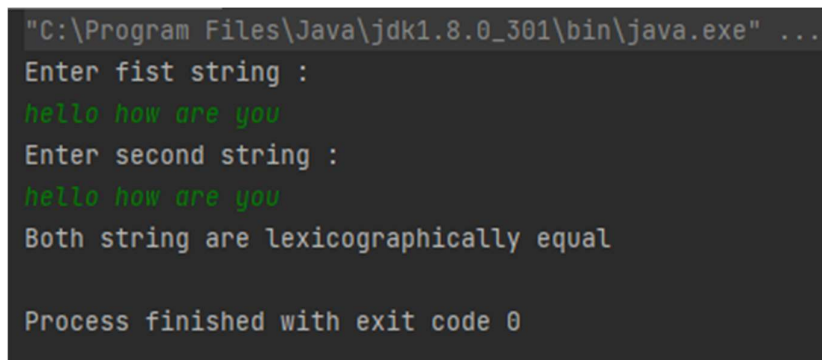
            System.out.println("Both string are lexicographically
equal ");

        else

            System.out.println("Both string are not equal ");

    }

}
```



```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
Enter first string :
hello how are you
Enter second string :
hello how are you
Both string are lexicographically equal

Process finished with exit code 0
```

6. Write a Java program to find the maximum occurring character in a string

```
import java.util.Scanner;

public class FindCharMax {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter String : ");

        String str = sc.nextLine();
```

```
int count[] = new int[256];
```

```
int len = str.length();
```

```
for (int i=0; i<len; i++)
```

```
    count[str.charAt(i)]++;
```

```
int max = -1;
```

```
char result = ' ';
```

```
for (int i = 0; i < len; i++) {
```

```
    if (max < count[str.charAt(i)]) {
```

```
        max = count[str.charAt(i)];
```

```
        result = str.charAt(i);
```

```
    }
```

```
}
```

```
    System.out.println("Maximum occurring character in a  
\""+str+"\" is '"+result+"'");
```

```
}
```

```
}
```

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...  
Enter String :  
hello how are you  
Maximum occurring character in a "hello how are you" is 'o'  
  
Process finished with exit code 0
```

**7. Write a Java program to check whether the first two characters present at the end of a given string.**

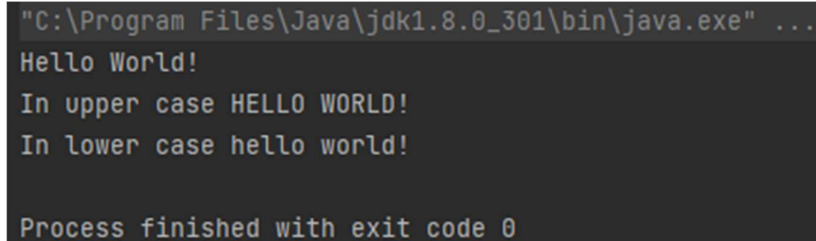
```
import java.util.Scanner;  
  
public class ComapareChar {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a string : ");  
        String str = sc.nextLine();  
  
        if (str.charAt(0) == str.charAt(str.length()-1) &&  
            str.charAt(1) == str.charAt(str.length()-2))  
            System.out.println("Yes");  
        else  
            System.out.println("No");  
    }  
}
```

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...  
Enter a string :  
how the ohhhhhhoh  
Yes  
  
Process finished with exit code 0
```

**8. Write a Java program to convert all the characters in a string to lowercase and Uppercase.**

```
import java.util.Scanner;  
  
public class ChangeLowerCase {  
    public static void main(String[] args) {  
        StringBuffer str = new StringBuffer("Hello World!");  
        int len = str.length();  
        System.out.println(str);  
        for (int i=0; i<len; i++){  
            Character c= str.charAt(i);  
            if (Character.isLowerCase(c))  
                str.replace(i,i+1,Character.toUpperCase(c)+"");  
        }  
        System.out.println("In upper case " +str);  
        for (int i=0; i<len; i++){  
            Character c= str.charAt(i);  
            if (Character.isUpperCase(c))  
                str.replace(i,i+1,Character.toLowerCase(c)+"");  
        }  
    }  
}
```

```
        System.out.println("In lower case "+str);  
    }  
}
```



```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...  
Hello World!  
In upper case HELLO WORLD!  
In lower case hello world!  
  
Process finished with exit code 0
```

**9. Write a class named Employee that has the following fields:**

- **name.** The name field references a String object that holds the employee's name.
- **idNumber.** The idNumber is an int variable that holds the employee's ID number.

**department.** The department field references a String object that holds the name of the department where the employee works.

**position.** The position field references a String object that holds the employee's job title.

**The class should have the following constructors:**

**A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's name, employee's ID number, department, and position.**

**A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's name and ID number. The department and position fields should be assigned an empty string (). A no-arg constructor that assigns empty strings () to the name, department, and position fields, and 0 to the idNumber field.**

**Write appropriate mutator methods that store values in these fields and accessor methods that return the values in these fields. Once you have written the class, write a separate program that creates three Employee objects to hold the following data: The program**

Name	ID Number	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

**should store this data in the three objects and then display the data for each employee on the screen.**

```
import java.util.Scanner;

public class Employee {
    int id;
    String name;
    String department;
    String position;

    public Employee() {
        this.name = null;
        this.id = 0;
        this.department = null;
        this.position = null;
    }
}
```

```
public Employee(int id, String name, String department,  
String position) {  
    this.id = id;  
    this.name = name;  
    this.department = department;  
    this.position = position;  
}
```

```
public int getId() {  
    return id;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public String getDepartment() {  
    return department;  
}
```

```
public String getPosition() {  
    return position;  
}
```



```
public static void displayEmployee(Employee[]
employees){

    System.out.println("Name\t\t ID Number\t\t
Department\t\t Position");

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

    for (Employee employee : employees){

        System.out.print(employee.getName());

        System.out.print("\t\t "+employee.getId());

    System.out.print("\t\t\t"+employee.getDepartment());

    System.out.println("\t\t\t"+employee.getPosition());

    }

}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter no of employee : ");

    int size = Integer.parseInt(sc.nextLine());

    Employee[] employee = new Employee[size];

    for (int i=0; i<size; i++){
```

```
System.out.println("Enter "+(i+1)+" Employee
details : ");

System.out.print("Enter Name of the employee : ");

String name = sc.nextLine();

System.out.print("Enter id of the employee : ");

int id = Integer.parseInt(sc.nextLine());

System.out.print("Enter department of the
employee : ");

String department = sc.nextLine();

System.out.print("Enter position of the employee :
");

String position = sc.nextLine();

employee[i] = new
Employee(id,name,department,position);
}

displayEmployee(employee);
}
}
```

Name	ID Number	Department	Position
=====			
HORIL PRASAD	123	CSED	Student
SANDEEP KUMAR	234	CSED	Student
GIRISH KUTTAN	345	CSED	Student
VIPIN KUMAR YADAV	456	CSED	Student
Process finished with exit code 0			

**10. Write a class named Car that has the following fields:**

- **yearModel.** The yearModel field is an int that holds the car's year model.
- **make.** The make field references a String object that holds the make of the car.
- **speed.** The speed field is an int that holds the car's current speed.

**In addition, the class should have the following constructor and other methods.**

**Constructor.** The constructor should accept the car's year model and make as arguments. These values should be assigned to the object's yearModel and make fields. The constructor should also assign 0 to the speed field.

**Accessors.** Appropriate accessor methods should get the values stored in an object's yearModel, make, and speed fields.

```
import java.util.Scanner;
```

```
public class Car {
```

```
    int yearModel;
```

```
    String make;
```

```
    int speed;
```

```
    public Car(int yearModel, String make, int speed) {
```

```
        this.yearModel = yearModel;
```

```
        this.make = make;
```

```
        this.speed = speed;  
    }
```

```
    public int getYearModel() {  
        return yearModel;  
    }
```

```
    public String getMake() {  
        return make;  
    }
```

```
    public int getSpeed() {  
        return speed;  
    }
```

```
    public static void displayCar(Car[] cars)  
    {  
        System.out.println("Year Model\t\t\t Maker\t\t\t  
Speed");  
  
        System.out.println("=====  
=====");  
  
        for (Car car : cars){  
            System.out.print(car.getYearModel());  
            System.out.print("\t\t\t"+car.getMake());
```

```
        System.out.print("\t\t\t"+car.getSpeed());

    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter no of cars : ");
    int size = Integer.parseInt(sc.nextLine());
    Car[] cars = new Car[size];
    for (int i=0; i<size; i++){
        System.out.println("Enter "+(i+1)+" car details : ");
        System.out.print("Enter year model of the car : ");
        int yearModel = Integer.parseInt(sc.nextLine());
        System.out.print("Enter make of the car : ");
        String make = sc.nextLine();
        System.out.print("Enter speed of the car : ");
        int speed = Integer.parseInt(sc.nextLine());

        cars[i] = new Car(yearModel,make,speed);

    }

    displayCar(cars);
}
```

```
}
```

Year	Model	Maker	Speed
=====			
2018		HONDA	140
2020		TOYOTA	180
2020		RENAULT	200
2022		BMW	240
2021		BUKATI	300

Process finished with exit code 0

**11.Design a class Personal Information that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator methods. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends' or family members' information.**

```
import java.util.Scanner;
```

```
public class PersonalInformation {
```

```
    String name;
```

```
    String address;
```

```
    int age;
```

```
    String phoneNumber;
```

```
    public PersonalInformation(String name, String address, int age, String  
phoneNumber) {
```

```
        this.name = name;
```

```
        this.address = address;
```

```
        this.age = age;
```

```
        this.phoneNumber = phoneNumber;
```

```
}
```

```
public String getAddress() {  
    return address;  
}
```

```
public void setAddress(String address) {  
    this.address = address;  
}
```

```
public String getName() {  
    return name;  
}
```

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

```
public int getAge() {  
    return age;  
}
```

```
public void setAge(int age) {  
    this.age = age;  
}
```

```
public String getPhoneNumber() {  
    return phoneNumber;  
}
```

```
public void setPhoneNumber(String phoneNumber) {
    this.phoneNumber = phoneNumber;
}

public static void displayInformation(PersonalInformation[]
personalInformations) {
    System.out.println("Name\t\t Address\t\t Age\t\t Phone Number");

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

    for (PersonalInformation personalInformation : personalInformations) {
        System.out.print(personalInformation.getName());
        System.out.print("\t\t " + personalInformation.getAddress());
        System.out.print("\t\t\t" + personalInformation.getAge());
        System.out.println("\t\t\t" + personalInformation.getPhoneNumber());

    }
}

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

    PersonalInformation[] personalInformation = new
PersonalInformation[3];

    for (int i = 0; i < 3; i++) {
        System.out.println("Enter name : ");
        String name = sc.nextLine();

        System.out.println("Enter address : ");
        String address = sc.nextLine();

        System.out.println("Enter age : ");
        int age = Integer.parseInt(sc.nextLine());
```



```
System.out.println("Enter phone number : ");

String phoneNumber = sc.nextLine();

    personalInformation[i] = new
    PersonallInformation(name,address,age,phoneNumber);

}

    displayInformation(personallInformation);

}

}
```

Name	Address	Age	Phone Number
=====	=====	=====	=====
Horil Prasad	GGDVGDDHGDVD	23	5757839928
SANDEEP KUMAR	JGDFDGFDFGF	24	6353656435
HHDVVDHGDVV	DGDVFHBBBHF	25	3765454433
Process finished with exit code 0			

12. Write a Circle class that has the following fields:

- radius: a double
  - PI: a final double initialized with the value 3.14159 The class should have the following methods.
  - Constructor. Accepts the radius of the circle as an argument.
  - Constructor. A no-arg constructor that sets the radius field to 0.0.
  - setRadius. A mutator method for the radius field.
  - getRadius. An accessor method for the radius field.
- getArea. Returns the area of the circle, which is calculated as  $\text{area} = \text{PI} * \text{radius} * \text{radius}$

**getDiameter.** Returns the diameter of the circle, which is calculated as  $\text{diameter} = \text{radius} * 2$

**getCircumference.** Returns the circumference of the circle, which is calculated as  $\text{circumference} = 2 * \text{PI} * \text{radius}$

Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

```
import java.util.Scanner;

public class Circle {
    double radius;
    final double PI = 3.14159;

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public Circle(double radius) {
        this.radius = radius;
    }

    public double getArea(double radius)
    {
        return PI*radius*radius;
    }

    public double getDiameter(double radius){
        return 2*radius;
    }
}
```

```
}

public double getCircumference(double radius){
    return 2*PI*radius;
}

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

    System.out.println("Enter radius of circle: ");
    double radius = sc.nextDouble();

    Circle circle = new Circle(radius);

    System.out.println("Area : "+circle.getArea(radius));
    System.out.println("Diameter: "+circle.getDiameter(radius));
    System.out.println("Circumference : "+circle.getCircumference(radius));
}
}
```

```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
Enter radius of circle:
34.65
Area : 3771.8636397749992
Diameter: 69.3
Circumference : 217.71218699999997

Process finished with exit code 0
```