

**Name: Mohd Hamza Abbasi**

**PRN: 22070122125**

**Batch: SY CSE B**

### **Questions:-**

Create a class NumberConverter with methods to convert between different number types (int, double, etc.). Include methods for converting a String to an Integer, a Double to a String, etc. Use wrapper classes for these conversions.

```
import java.util.Scanner;

public class first{

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        numberconverter obj=new numberconverter();

        System.out.print("Enter an Integer:");

        int a=sc.nextInt();

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

        Double b=sc.nextDouble();

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

        String s1=sc.nextLine();

        String s=sc.nextLine();

        System.out.println(obj.stringToInteger(s));

        System.out.println(obj.doubleToString(b));

        System.out.println(obj.IntegertoString(a));

        System.out.println(obj.doubleFromString(s));

    }

}

class numberconverter{

    public Integer stringToInteger(String str) {

        try {

            return Integer.valueOf(str);

        } catch (NumberFormatException e) {

            System.err.println("Error converting string to integer: " + e.getMessage());

        }

    }

}
```

```

        return null;
    }
}

public Double doubleFromString(String str) {
    try {
        return Double.valueOf(str);
    } catch (NumberFormatException e) {
        System.err.println("Error converting string to double: " + e.getMessage());
        return null;
    }
}

public String doubleToString(Double d) {
    return String.valueOf(d);
}

public String IntegertoString(Integer a){
    return String.valueOf(a);
}
}

```

Create a StatisticsCalculator class with methods to calculate statistics on a list of numbers. Include methods to find the minimum, maximum, average, and standard deviation. Use wrapper classes to handle numeric values.

```

import java.util.Scanner;

public class first{

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

        statisticsCalculator obj=new statisticsCalculator();

        System.out.print("Enter the count of numbers required:");

        int a=sc.nextInt();

        int arr[]=new int[a];
    }
}

```

```

        for(int j=0;j<a;j++){
            arr[j]=sc.nextInt();
        }
        obj.findminimum(arr,a);
        obj.findmaximum(arr,a);
        obj.findaverage(arr, a);
        obj.findstandarddeviation(arr,a);
        sc.close();
    }
}

class statisticsCalculator{
    public void findminimum(int arr1[],int length){
        int min=122323;
        for(int j=0;j<length;j++){
            if(arr1[j]<min){
                min=arr1[j];
            }
        }
        System.out.println("The minimum number is "+min);
    }
    public void findmaximum(int arr1[],int length){
        int max=-23;
        for(int j=0;j<length;j++){
            if(arr1[j]>max){
                max=arr1[j];
            }
        }
        System.out.println("The maximum number is "+max);
    }
    public void findaverage(int arr1[],int length){
        int sum=0;
        for(int j=0;j<length;j++){
            sum=sum+arr1[j];
        }
    }
}

```

```

    }

    System.out.println("The average of the numbers is "+Double.valueOf(sum/length));
}

public void findstandarddeviation(int arr1[],int length){

    int sum=0;

    for(int j=0;j<length;j++){

        sum=sum+arr1[j];

    }

    Double average=Double.valueOf(sum/length);

    Double sd=Double.valueOf(0);

    for(int j=0;j<length;j++){

        sd=sd+(arr1[j]-average)*(arr1[j]-average);

    }

    System.out.println("The standard deviation of the numbers is "+Double.valueOf(sd/length));

}

}

```

Implement an online shopping system with a base class Product and subclasses like Electronics, Clothing, and Books. Each subclass should have specific properties/methods.

```

class Product {

    public String name;

    public double price;

    public Product(String name, double price) {

        this.name = name;

        this.price = price;

    }

    public String getName() {

        return name;

    }

}

```

```
    public double getPrice() {  
        return price;  
    }  
}
```

```
class Electronics extends Product {  
    public String brand;  
  
    public Electronics(String name, double price, String brand) {  
        super(name, price);  
        this.brand = brand;  
    }  
  
    public String getBrand() {  
        return brand;  
    }  
}
```

```
class Clothing extends Product {  
    public String size;  
  
    public Clothing(String name, double price, String size) {  
        super(name, price);  
        this.size = size;  
    }  
  
    public String getSize() {  
        return size;  
    }  
}
```

```
class Book extends Product {  
    public String author;
```

```

public Book(String name, double price, String author) {
    super(name, price);
    this.author = author;
}

public String getAuthor() {
    return author;
}
}

public class onlineshoppingsystem {
    public static void main(String[] args) {
        Electronics laptop = new Electronics("Laptop", 999.99, "Apple");
        Clothing shirt = new Clothing("T-Shirt", 19.99, "Large");
        Book novel = new Book("To Kill a Mockingbird", 12.99, "Harper Lee");

        System.out.println("Product: " + laptop.getName() + ", Brand: " + laptop.getBrand() + ", Price: $" +
laptop.getPrice());

        System.out.println("Product: " + shirt.getName() + ", Size: " + shirt.getSize() + ", Price: $" + shirt.getPrice());

        System.out.println("Product: " + novel.getName() + ", Author: " + novel.getAuthor() + ", Price: $" +
novel.getPrice());
    }
}

```

Design a library management system with a base class `LibraryItem` and subclasses for different types of items like `Book`, `DVD`, and `Magazine`. Each subclass should have specific properties/methods.

```

class LibraryItem {
    public String title;
    public String author;

    public LibraryItem(String title, String author) {
        this.title = title;
    }
}

```

```
        this.author = author;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }
}

class Book extends LibraryItem {
    public int pageCount;

    public Book(String title, String author, int pageCount) {
        super(title, author);
        this.pageCount = pageCount;
    }

    public int getPageCount() {
        return pageCount;
    }
}

class DVD extends LibraryItem {
    public int duration; // in minutes

    public DVD(String title, String author, int duration) {
        super(title, author);
        this.duration = duration;
    }
}
```

```

    public int getDuration() {
        return duration;
    }
}

```

```

class Magazine extends LibraryItem {
    public int issueNumber;

    public Magazine(String title, String author, int issueNumber) {
        super(title, author);
        this.issueNumber = issueNumber;
    }
}

```

```

    public int getIssueNumber() {
        return issueNumber;
    }
}

```

```

public class librarymanagementsystem {
    public static void main(String[] args) {
        Book book = new Book("To Kill a Mockingbird", "Harper Lee", 281);
        DVD dvd = new DVD("Inception", "Christopher Nolan", 148);
        Magazine magazine = new Magazine("National Geographic", "Various Authors", 123);

        System.out.println("Item: " + book.getTitle() + ", Author: " + book.getAuthor() + ", Pages: " +
            book.getPageCount());

        System.out.println("Item: " + dvd.getTitle() + ", Author: " + dvd.getAuthor() + ", Duration: " +
            dvd.getDuration() + " minutes");

        System.out.println("Item: " + magazine.getTitle() + ", Author: " + magazine.getAuthor() + ", Issue Number: " +
            magazine.getIssueNumber());
    }
}

```

Create a class DistanceCalculator with methods to calculate distance in various units. Implement overloaded methods for calculating distance in meters and kilometers. Also, provide methods for distance between two points given their coordinates.



```

public class distancecalculator {

    public static double calculateDistanceInMeters(double lat1, double lon1, double lat2, double lon2) {

        double earthRadius = 6371000;

        double dLat = Math.toRadians(lat2 - lat1);

        double dLon = Math.toRadians(lon2 - lon1);

        double a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +

            Math.cos(Math.toRadians(lat1)) * Math.cos(Math.toRadians(lat2)) *

            Math.sin(dLon / 2) * Math.sin(dLon / 2);

        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

        double distance = earthRadius * c;

        return distance;

    }

    public static double calculateDistanceInKilometers(double lat1, double lon1, double lat2, double lon2) {

        double distanceInMeters = calculateDistanceInMeters(lat1, lon1, lat2, lon2);

        double distanceInKilometers = distanceInMeters / 1000;

        return distanceInKilometers;

    }

    public static void main(String[] args) {

        double lat1 = 40.7128;

        double lon1 = -74.0060;

        double lat2 = 34.0522;

        double lon2 = -118.2437;

        double distanceInMeters = calculateDistanceInMeters(lat1, lon1, lat2, lon2);

        System.out.println("Distance between New York City and Los Angeles in meters: " + distanceInMeters +
            "m");

        double distanceInKilometers = calculateDistanceInKilometers(lat1, lon1, lat2, lon2);

        System.out.println("Distance between New York City and Los Angeles in kilometers: " +
            distanceInKilometers + "km");

    }
}

```

```
}
```

Design a class `SearchEngine` with overloaded methods for searching. Include methods for searching by keyword, by date range, and by relevance. The methods should handle different data types and search criteria.

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
class Searchable<T> {
```

```
    private T data;
```

```
    private String keyword;
```

```
    private String date;
```

```
    public Searchable(T data, String keyword, String date) {
```

```
        this.data = data;
```

```
        this.keyword = keyword;
```

```
        this.date = date;
```

```
    }
```

```
    public T getData() {
```

```
        return data;
```

```
    }
```

```
    public String getKeyword() {
```

```
        return keyword;
```

```
    }
```

```
    public String getDate() {
```

```
        return date;
```

```
    }
```

```
}
```

```
public class SearchEngine {
```

```

public static <T> List<T> searchByKeyword(String keyword, List<Searchable<T>> data) {

    List<T> results = new ArrayList<>();

    for (Searchable<T> item : data) {

        if (item.getKeyword().toLowerCase().contains(keyword.toLowerCase())) {

            results.add(item.getData());

        }

    }

    return results;

}

```

```

public static <T> List<T> searchByDateRange(String startDate, String endDate, List<Searchable<T>> data) {

    List<T> results = new ArrayList<>();

    for (Searchable<T> item : data) {

        if (isWithinDateRange(item.getDate(), startDate, endDate)) {

            results.add(item.getData());

        }

    }

    return results;

}

```

```

public static <T> List<T> searchByRelevance(String query, List<Searchable<T>> data) {

    List<T> results = new ArrayList<>();

    for (Searchable<T> item : data) {

        if (item.getKeyword().toLowerCase().contains(query.toLowerCase())) {

            results.add(item.getData());

        }

    }

    return results;

}

```

```

public static void main(String[] args) {

    List<Searchable<String>> stringData = new ArrayList<>();

    stringData.add(new Searchable<>("Document 1", "keyword1", "2024-01-15"));

}

```

```

stringData.add(new Searchable<>("Document 2", "keyword2", "2024-01-20"));

List<Searchable<Integer>> intData = new ArrayList<>();
intData.add(new Searchable<>(1, "keyword1", "2024-01-15"));
intData.add(new Searchable<>(2, "keyword2", "2024-01-20"));

List<String> stringKeywordResults = SearchEngine.searchByKeyword("keyword1", stringData);
System.out.println("Search by keyword for string data: " + stringKeywordResults);

List<Integer> intKeywordResults = SearchEngine.searchByKeyword("keyword2", intData);
System.out.println("Search by keyword for integer data: " + intKeywordResults);

List<String> stringDateResults = SearchEngine.searchByDateRange("2024-01-15", "2024-01-20",
stringData);
System.out.println("Search by date range for string data: " + stringDateResults);

List<Integer> intDateResults = SearchEngine.searchByDateRange("2024-01-15", "2024-01-20", intData);
System.out.println("Search by date range for integer data: " + intDateResults);

List<String> stringRelevanceResults = SearchEngine.searchByRelevance("keyword1", stringData);
System.out.println("Search by relevance for string data: " + stringRelevanceResults);

List<Integer> intRelevanceResults = SearchEngine.searchByRelevance("keyword2", intData);
System.out.println("Search by relevance for integer data: " + intRelevanceResults);
}

private static boolean isWithinDateRange(String date, String startDate, String endDate) {
    return date.compareTo(startDate) >= 0 && date.compareTo(endDate) <= 0;
}
}

```

Develop a base class Post with a method publish(). Derive subclasses TextPost, ImagePost, and VideoPost. Override the publish() method in each subclass to handle the specifics of publishing each type of post.

```

class Post {

```

```
public String content;

public Post(String content) {
    this.content = content;
}

public void publish() {
    System.out.println("Publishing generic post: " + content);
}
}

class TextPost extends Post {
    public String title;

    public TextPost(String title, String content) {
        super(content);
        this.title = title;
    }

    @Override
    public void publish() {
        System.out.println("Publishing text post titled '" + title + "': " + content);
    }
}

class ImagePost extends Post {
    public String imageUrl;

    public ImagePost(String imageUrl, String content) {
        super(content);
        this.imageUrl = imageUrl;
    }
}
```

```

@Override

public void publish() {
    System.out.println("Publishing image post with URL '" + imageUrl + "': " + content);
}
}

```

```

class VideoPost extends Post {
    public String videoUrl;

    public VideoPost(String videoUrl, String content) {
        super(content);
        this.videoUrl = videoUrl;
    }
}

```

```

@Override

public void publish() {
    System.out.println("Publishing video post with URL '" + videoUrl + "': " + content);
}
}

```

```

public class First {
    public static void main(String[] args) {
        // Example usage

        TextPost textPost = new TextPost("Important Update", "Here's an important update regarding our latest project.");
        textPost.publish();

        ImagePost imagePost = new ImagePost("https://example.com/image.jpg", "Check out this beautiful scenery!");
        imagePost.publish();

        VideoPost videoPost = new VideoPost("https://example.com/video.mp4", "Watch this amazing video!");
        videoPost.publish();
    }
}

```

```
}
```

Develop a base class Beverage with methods prepare() and serve(). Derive subclasses Coffee, Tea, and Smoothie. Override the methods in each subclass to simulate the preparation and serving of different beverages.

```
class Beverage {  
    public void prepare() {  
        System.out.println("Preparing generic beverage...");  
    }  
  
    public void serve() {  
        System.out.println("Serving generic beverage...");  
    }  
}
```

```
class Coffee extends Beverage {  
    public void prepare() {  
        System.out.println("Brewing coffee...");  
    }  
  
    public void serve() {  
        System.out.println("Serving coffee in a mug.");  
    }  
}
```

```
class Tea extends Beverage {  
    public void prepare() {  
        System.out.println("Steeping tea...");  
    }  
  
    public void serve() {  
        System.out.println("Serving tea in a teacup.");  
    }  
}
```

```
}
```

```
class Smoothie extends Beverage {
```

```
    public void prepare() {
```

```
        System.out.println("Blending smoothie...");
```

```
    }
```

```
    public void serve() {
```

```
        System.out.println("Serving smoothie in a glass.");
```

```
    }
```

```
}
```

```
public class first {
```

```
    public static void main(String[] args) {
```

```
        Coffee coffee = new Coffee();
```

```
        coffee.prepare();
```

```
        coffee.serve();
```

```
        System.out.println();
```

```
        Tea tea = new Tea();
```

```
        tea.prepare();
```

```
        tea.serve();
```

```
        System.out.println();
```

```
        Smoothie smoothie = new Smoothie();
```

```
        smoothie.prepare();
```

```
        smoothie.serve();
```

```
    }
```

```
}
```