JAVA

Task 1 :

Write a Java program that performs the following operations on a given string: find its length, convert it to uppercase, extract a substring, and replace a character.

CODE :

```
import java.util.Scanner;

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

       System.out.print("Enter a string: ");
       String string = scanner.nextLine();

       int length = string.length();
       System.out.println("Length : " + length);

       String uppercase = string.toUpperCase();
       System.out.println("Uppercase string: " + uppercase);

       System.out.print("Enter index (for substring):");
       int number = scanner.nextInt();

       if (length >= number) {
           String substring = string.substring(number);
           System.out.println("Substring: " + substring);
       } else {
           System.out.println("Index out of range");
```

```
        }

        System.out.print("Enter the character to be replaced: ");
        char x = scanner.next().charAt(0);
        System.out.print("Enter the character to replace with: ");
        char y = scanner.next().charAt(0);

        if (string.indexOf(x) != -1) {
            String new_string = string.replace(x, y);
            System.out.println("New string: " + new_string);
        } else {
            System.out.println("Character not in the string.");
        }

        scanner.close();
    }
}
```

OUTPUT :

Enter a string: stare
Length : 5
Uppercase string: STARE
Enter index (for substring):3
Substring: re
Enter the character to be replaced: e
Enter the character to replace with: s
New string: stars

Task 2 :

Write a Java program to parse a string into different primitive data types using wrapper class methods like parseInt, parseDouble, parseBoolean, etc., and convert primitive types to strings using valueOf

CODE :

```
import java.util.Scanner;

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

       System.out.print("Integer string: ");
       String int_str = scanner.nextLine();
       System.out.print("Double string: ");
       String double_str = scanner.nextLine();
       System.out.print("Boolean string: ");
       String boolean_str = scanner.nextLine();

       int parsed_int = Integer.parseInt(int_str);
       double parsed_double = Double.parseDouble(double_str);
       boolean parsed_boolean = Boolean.parseBoolean(boolean_str);

       System.out.println("Parsed integer: " + parsed_int);
       System.out.println("Parsed double: " + parsed_double);
       System.out.println("Parsed boolean: " + parsed_boolean);

       String int_to_str = String.valueOf(parsed_int);
       String double_to_str = String.valueOf(parsed_double);
       String boolean_to_str = String.valueOf(parsed_boolean);
```

```
        System.out.println("Integer to string: " + int_to_str);
        System.out.println("Double to string: " + double_to_str);
        System.out.println("Boolean to string: " + boolean_to_str);

        scanner.close();
    }
}
```

OUTPUT :

Integer string: 98765
Double string: 987.65
Boolean string: true
Parsed integer: 98765
Parsed double: 987.65
Parsed boolean: true
Integer to string: 98765
Double to string: 987.65
Boolean to string: true

Task 3 :

Write a Java program to sort an array of integers in ascending order using a sorting algorithm of your choice

CODE :

```java
import java.util.Scanner;

public class InsertionSort {

   public static void insertionSort(int[] arr,int n) {

       if (n>1){

           for (int i = 1; i < n; i++) {
               int key = arr[i];
               int j = i - 1;

               while (j >= 0 && arr[j] > key) {
                   arr[j + 1] = arr[j];
                   j = j - 1;
               }
               arr[j + 1] = key;
           }
       }
   }

   public static void print_arr(int[] array) {
       for (int value : array) {
           System.out.print(value + " ");
       }
       System.out.println();
   }

   public static void main(String[] args) {

       Scanner scanner = new Scanner(System.in);

       System.out.print("Enter the number of elements : ");
       int n = scanner.nextInt();

       int[] arr = new int[n];
       System.out.println("Enter the elements :");
       for (int i = 0; i < n; i++) {
           arr[i] = scanner.nextInt();
```

```
        }

        System.out.println("Array:");
        print_arr(arr);

        insertionSort(arr,n);

        System.out.println("Array sorted:");
        print_arr(arr);
    }
}
```

OUTPUT :

Enter the number of elements : 8
Enter the elements :
98
12
76
43
45
87
56
11
Array:98 12 76 43 45 87 56 11
Array sorted:11 12 43 45 56 76 87 98

Task 4 :

Use an ArrayList to store the list of books. Each book should have attributes such as title, author, ISBN, and price.Implement functionalities to add new books, remove existing books, and display all books in the library.

CODE :

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

class Book {
   private String title;
   private String author;
   private String isbn;
   private double price;

   private static ArrayList<Book> books = new ArrayList<>();

   public Book(String title, String author, String isbn, double price) {
       this.title = title;
       this.author = author;
       this.isbn = isbn;
       this.price = price;
   }

   public String getIsbn() {
       return isbn;
   }

   @Override
   public String toString() {
```

```
        return "Title: " + title + ", Author: " + author + ", ISBN: " +
isbn + ", Price: $" + price;
    }

    public static void add_book(Book book) {
        books.add(book);
        System.out.println("Book added.");
    }

    public static void rem_book(String isbn) {
        for (int i = 0; i < books.size(); i++) {
            if (books.get(i).getIsbn()==isbn) {
                books.remove(i);
                System.out.println("Book removed.");
                return;
            }
        }
        System.out.println("Book with ISBN " + isbn + " not found.");
    }

    public static void dis_books() {
        if (books.isEmpty()) {
            System.out.println("No books");
        } else {
            for (Book book : books) {
                System.out.println(book);
            }
        }
    }

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

        do {

            System.out.println("1. Add a Book");
            System.out.println("2. Remove a Book");
            System.out.println("3. Display all Books");
            System.out.println("4. Exit");
```

```
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter title: ");
                    String title = scanner.nextLine();
                    System.out.print("Enter name of the author: ");
                    String author = scanner.nextLine();
                    System.out.print("Enter ISBN: ");
                    String isbn = scanner.nextLine();
                    System.out.print("Enter price: ");
                    double price = scanner.nextDouble();
                    scanner.nextLine();
                    Book new_book = new Book(title, author, isbn, price);
                    Book.add_book(new_book);
                    break;

                case 2:
                    System.out.print("Enter ISBN of the book to remove: ");
                    String removeIsbn = scanner.nextLine();
                    Book.rem_book(removeIsbn);
                    break;

                case 3:
                    Book.dis_books();
                    break;

                default:
                    System.out.println("Invalid choice");
            }
        } while (choice != 4);

        scanner.close();
    }
}
```

OUTPUT :

1. Add a Book
2. Remove a Book
3. Display all Books
4. Exit
Enter your choice: 1
Enter title: The God of Small Things
Enter name of the author: Arundathi Roy
Enter ISBN: 987654321
Enter price: 12
Book added.
1. Add a Book
2. Remove a Book
3. Display all Books
4. Exit
Enter your choice: 3
Title: The God of Small Things, Author: Arundathi Roy, ISBN: 987654321, Price: $12.0
1. Add a Book
2. Remove a Book
3. Display all Books
4. Exit
Enter your choice: 2
Enter ISBN of the book to remove: 987654321
Book removed.
1. Add a Book
2. Remove a Book

3. Display all Books
4. Exit
Enter your choice: 3
No books
1. Add a Book
2. Remove a Book
3. Display all Books
4. Exit
Enter your choice: 4

Task 5 :

Use a HashSet to manage the unique genres available in the library.Ensure that new genres can be added without duplicating existing genres

CODE :

```
import java.util.ArrayList;
import java.util.Scanner;
import java.util.HashSet;

class Book {
   private String title;
   private String author;
   private String isbn;
   private double price;
   private String genre;
```

```
    private static ArrayList<Book> books = new ArrayList<>();
    private static HashSet<String> genres = new HashSet<>();

    public Book(String title, String author, String isbn, double
price,String genre) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.price = price;
        this.genre = genre;
        genres.add(genre);
    }

    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public String getIsbn() { return isbn; }
    public double getPrice() { return price; }
    public String getGenre() { return genre; }

    @Override
    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " +
isbn + ", Price: $" + price + ", Genre: " + genre;
    }

    public static void add_book(Book book) {
        books.add(book);
        System.out.println("Book added.");
    }

    public static void rem_book(String isbn) {
        for (int i = 0; i < books.size(); i++) {
            if (books.get(i).getIsbn().equals(isbn)) {
                genres.remove(books.get(i).getGenre());
                books.remove(i);
                System.out.println("Book removed.");
                return;
            }
        }
        System.out.println("Book with ISBN " + isbn + " not found.");
```

```
    }

    public static void dis_books() {
        if (books.isEmpty()) {
            System.out.println("No books");
        } else {
            for (Book book : books) {
                System.out.println(book);
            }
        }
    }

    public static void add_genre(String genre) {
        if (genres.add(genre)) {
            System.out.println("Genre added.");
        } else {
            System.out.println("Genre already exists.");
        }
    }

    public static void dis_genres() {
        if (genres.isEmpty()) {
            System.out.println("No genres.");
        } else {
            for (String genre : genres) {
                System.out.println(genre);
            }
        }
    }

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

        do {

            System.out.println("1. Add a Book");
            System.out.println("2. Remove a Book");
            System.out.println("3. Display all Books");
            System.out.println("4. Add Genre");
```

```
            System.out.println("5. Display all Genres");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter title: ");
                    String title = scanner.nextLine();
                    System.out.print("Enter author: ");
                    String author = scanner.nextLine();
                    System.out.print("Enter ISBN: ");
                    String isbn = scanner.nextLine();
                    System.out.print("Enter price: ");
                    double price = scanner.nextDouble();
                    scanner.nextLine();
                    System.out.print("Enter genre: ");
                    String genre = scanner.nextLine();
                    Book new_book = new Book(title, author, isbn, price,
genre);
                    Book.add_book(new_book);
                    break;

                case 2:
                    System.out.print("Enter ISBN of the book to remove: ");
                    String remove_isbn = scanner.nextLine();
                    Book.rem_book(remove_isbn);
                    break;

                case 3:
                    Book.dis_books();
                    break;

                case 4:
                    System.out.print("Enter genre: ");
                    String new_genre = scanner.nextLine();
                    Book.add_genre(new_genre);
                    break;
```

```
                case 5:
                    Book.dis_genres();
                    break;

                case 6:
                    break;


                default:
                    System.out.println("Invalid choice");
            }
        } while (choice != 6);

        scanner.close();
    }
}
```

OUTPUT :

1. Add a Book
2. Remove a Book
3. Display all Books
4. Add Genre
5. Display all Genres
6. Exit

Enter your choice: 1
Enter title: Dracula
Enter author: Bram Stoker
Enter ISBN: 987654321
Enter price: 14

Enter genre: Fiction
Book added.
1. Add a Book
2. Remove a Book
3. Display all Books
4. Add Genre
5. Display all Genres
6. Exit
Enter your choice: 4
Enter genre: Thriller
Genre added.
1. Add a Book
2. Remove a Book
3. Display all Books
4. Add Genre
5. Display all Genres
6. Exit
Enter your choice: 5
Thriller
Fiction
1. Add a Book
2. Remove a Book
3. Display all Books
4. Add Genre
5. Display all Genres
6. Exit
Enter your choice: 6

Task 6 :

Use a HashMap to map ISBN numbers to books for quick lookup.Implement functionalities to add, update, and retrieve book details using ISBN.

CODE :

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Scanner;

class Book {
   private String title;
   private String author;
   private String isbn;
   private double price;
   private String genre;

   private static ArrayList<Book> books = new ArrayList<>();
   private static HashSet<String> genres = new HashSet<>();
   private static Map<String, Book> bookMap = new HashMap<>();

   public Book(String title, String author, String isbn, double price,
String genre) {
       this.title = title;
       this.author = author;
       this.isbn = isbn;
       this.price = price;
       this.genre = genre;
       genres.add(genre);
   }
```

```
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public String getIsbn() { return isbn; }
    public double getPrice() { return price; }
    public String getGenre() { return genre; }

    @Override
    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " +
isbn + ", Price: $" + price + ", Genre: " + genre;
    }

    public static void add_book(Book book) {
        if (bookMap.putIfAbsent(book.getIsbn(), book) == null) {
            books.add(book);
            System.out.println("Book added.");
        } else {
            System.out.println("Book with ISBN " + book.getIsbn() + "
already exists.");
        }
    }

    public static void rem_book(String isbn) {
        Book book = bookMap.remove(isbn);
        if (book != null) {
            books.remove(book);
            boolean genreInUse = books.stream().anyMatch(b ->
b.getGenre().equals(book.getGenre()));
            if (!genreInUse) {
                genres.remove(book.getGenre());
            }
            System.out.println("Book removed.");
        } else {
            System.out.println("Book with ISBN " + isbn + " not found.");
        }
    }

    public static void update_book(String isbn, String title, String
author, double price, String genre) {
```

```
        Book existingBook = bookMap.get(isbn);
        if (existingBook != null) {
            existingBook.title = title;
            existingBook.author = author;
            existingBook.price = price;
            genres.remove(existingBook.getGenre());
            existingBook.genre = genre;
            genres.add(genre);
            System.out.println("Book updated.");
        } else {
            System.out.println("Book with ISBN " + isbn + " not found.");
        }
    }

    public static Book get_book(String isbn) {
        return bookMap.get(isbn);
    }

    public static void dis_books() {
        if (books.isEmpty()) {
            System.out.println("No books.");
        } else {
            for (Book book : books) {
                System.out.println(book);
            }
        }
    }

    public static void add_genre(String genre) {
        if (genres.add(genre)) {
            System.out.println("Genre added.");
        } else {
            System.out.println("Genre already exists.");
        }
    }

    public static void dis_genres() {
        if (genres.isEmpty()) {
            System.out.println("No genres.");
        } else {
```

```
            for (String genre : genres) {
                System.out.println(genre);
            }
        }
    }

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

        do {
            System.out.println("1. Add a Book");
            System.out.println("2. Remove a Book");
            System.out.println("3. Update a Book");
            System.out.println("4. Retrieve a Book");
            System.out.println("5. Display all Books");
            System.out.println("6. Add Genre");
            System.out.println("7. Display all Genres");
            System.out.println("8. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter title: ");
                    String title = scanner.nextLine();
                    System.out.print("Enter author: ");
                    String author = scanner.nextLine();
                    System.out.print("Enter ISBN: ");
                    String isbn = scanner.nextLine();
                    System.out.print("Enter price: ");
                    double price = scanner.nextDouble();
                    scanner.nextLine();
                    System.out.print("Enter genre: ");
                    String genre = scanner.nextLine();
                    Book new_book = new Book(title, author, isbn, price,
genre);
                    Book.add_book(new_book);
                    break;
```

```
                case 2:
                    System.out.print("Enter ISBN of the book to remove: ");
                    String remove_isbn = scanner.nextLine();
                    Book.rem_book(remove_isbn);
                    break;

                case 3:
                    System.out.print("Enter ISBN of the book to update: ");
                    String update_isbn = scanner.nextLine();
                    System.out.print("Enter new title: ");
                    String new_title = scanner.nextLine();
                    System.out.print("Enter new author: ");
                    String new_author = scanner.nextLine();
                    System.out.print("Enter new price: ");
                    double new_price = scanner.nextDouble();
                    scanner.nextLine();
                    System.out.print("Enter new genre: ");
                    String new_genre = scanner.nextLine();
                    Book.update_book(update_isbn, new_title, new_author,
new_price, new_genre);
                    break;

                case 4:
                    System.out.print("Enter ISBN of the book to retrieve:
");
                    String ret_isbn = scanner.nextLine();
                    Book book = Book.get_book(ret_isbn);
                    if (book != null) {
                        System.out.println("Book details: " + book);
                    } else {
                        System.out.println("Book with ISBN " + ret_isbn + "
not found.");
                    }
                    break;

                case 5:
                    Book.dis_books();
                    break;
```

```
                case 6:
                    System.out.print("Enter genre: ");
                    String genre_add = scanner.nextLine();
                    Book.add_genre(genre_add);
                    break;

                case 7:
                    Book.dis_genres();
                    break;

                case 8:
                    break;

                default:
                    System.out.println("Invalid choice");
            }
        } while (choice != 8);

        scanner.close();
    }
}
```

OUTPUT :

1. Add a Book
2. Remove a Book
3. Update a Book
4. Retrieve a Book
5. Display all Books
6. Add Genre
7. Display all Genres
8. Exit

Enter your choice: 1

Enter title: Some Name
Enter author: Author Name
Enter ISBN: 121
Enter price: 11
Enter genre: Fiction
Book added.
1. Add a Book
2. Remove a Book
3. Update a Book
4. Retrieve a Book
5. Display all Books
6. Add Genre
7. Display all Genres
8. Exit
Enter your choice: 3
Enter ISBN of the book to update: 121
Enter new title: New
Enter new author: Other Author
Enter new price: 12
Enter new genre: Adventure
Book updated.
1. Add a Book
2. Remove a Book
3. Update a Book
4. Retrieve a Book
5. Display all Books
6. Add Genre

7. Display all Genres
8. Exit
Enter your choice: 5
Title: New, Author: Other Author, ISBN: 121, Price: $12.0, Genre: Adventure
1. Add a Book
2. Remove a Book
3. Update a Book
4. Retrieve a Book
5. Display all Books
6. Add Genre
7. Display all Genres
8. Exit
Enter your choice: 4
Enter ISBN of the book to retrieve: 121
Book details: Title: New, Author: Other Author, ISBN: 121, Price: $12.0, Genre: Adventure
1. Add a Book
2. Remove a Book
3. Update a Book
4. Retrieve a Book
5. Display all Books
6. Add Genre
7. Display all Genres
8. Exit
Enter your choice: 8

Task 7 :

Implement a exception called ProductNotFoundException that is thrown when a product is not found in the inventory.Use try, catch, finally, throw, and throws to handle exceptions appropriately

CODE :

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

class ProductNotFoundException extends Exception {
    public ProductNotFoundException(String message) {
        super(message);
    }
}

class Product {
    private String name;
    private String id;
    private double price;

    public Product(String name, String id, double price) {
        this.name = name;
        this.id = id;
        this.price = price;
    }

    public String getName() { return name; }
    public String getId() { return id; }
    public double getPrice() { return price; }
```

```
    @Override
    public String toString() {
        return "Name: " + name + ", ID: " + id + ", Price: $" + price;
    }
}

class Inventory {
    private Map<String, Product> products = new HashMap<>();

    public Inventory() {

        products.put("100", new Product("Laptop", "100", 1000));
        products.put("101", new Product("Smartphone", "101", 600));
        products.put("102", new Product("Tablet", "102", 780));
    }

    public Product getProduct(String id) throws ProductNotFoundException {
        if (products.containsKey(id)) {
            return products.get(id);
        } else {
            throw new ProductNotFoundException("Product with ID " + id + "
not found.");
        }
    }
}

public class Main3 {
    public static void main(String[] args) {
        Inventory inventory = new Inventory();
        Scanner scanner = new Scanner(System.in);
        String id;

        System.out.print("Enter product ID: ");
        id = scanner.nextLine();

        try {
            Product product = inventory.getProduct(id);
            System.out.println("Product details: " + product);
        } catch (ProductNotFoundException e) {
            System.out.println(e.getMessage());
```

```
        } finally {
            System.out.println("The End");
        }

        scanner.close();
    }
}
```

OUTPUT :

Enter product ID: 101
Product details: Name: Smartphone, ID: 101, Price: $600.0
The End
Enter product ID: 105
Product with ID 105 not found.
The End

Task 8 :

Use any one of the file handling to read employee records from a text file and write employee records to a text file

CODE :

```
import java.io.*;
import java.util.ArrayList;
import java.util.List;
```

```java
class Employee {
    private String name;
    private String id;
    private double salary;

    public Employee(String name, String id, double salary) {
        this.name = name;
        this.id = id;
        this.salary = salary;
    }

    public String getName() { return name; }
    public String getId() { return id; }
    public double getSalary() { return salary; }

    @Override
    public String toString() {
        return name + "," + id + "," + salary;
    }

    public static Employee fromString(String record) {
        String[] parts = record.split(",");
        if (parts.length == 3) {
            String name = parts[0].trim();
            String id = parts[1].trim();
            double salary = Double.parseDouble(parts[2].trim());
            return new Employee(name, id, salary);
        }
        return null;
    }
}

public class Main4 {
    private static final String FILE_PATH = "employees.txt";

    public static void main(String[] args) {
        List<Employee> employees = new ArrayList<>();

        read(employees);
```

```
        display(employees);

        write(employees);
    }

    private static void read(List<Employee> employees) {
        try (BufferedReader br = new BufferedReader(new
FileReader(FILE_PATH))) {
            String line;
            while ((line = br.readLine()) != null) {
                Employee employee = Employee.fromString(line);
                if (employee != null) {
                    employees.add(employee);
                }
            }
            System.out.println("Employees loaded from file.");
        } catch (IOException e) {
            System.out.println("Error reading file: " + e.getMessage());
        }
    }

    private static void write(List<Employee> employees) {
        try (BufferedWriter bw = new BufferedWriter(new
FileWriter(FILE_PATH))) {
            for (Employee emp : employees) {
                bw.write(emp.toString());
                bw.newLine();
            }
            System.out.println("Employees saved to file.");
        } catch (IOException e) {
            System.out.println("Error writing file: " + e.getMessage());
        } finally {
            System.out.println("File operation completed.");
        }
    }

    private static void display(List<Employee> employees) {
        if (employees.isEmpty()) {
            System.out.println("No employees");
        } else {
```

```
            for (Employee emp : employees) {
                System.out.println(emp);
            }
        }
    }
}
```

OUTPUT :

Employees loaded from file.
James Austin, 100, 50000
Felix Smith, 101, 60000
Dylan Johnson, 102, 70000
Employees saved to file.
File operation completed.