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:

```java
import java.util.Scanner;

public class StringOperation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String st = sc.nextLine();

        System.out.println("The length of the string is " + st.length());
        System.out.println("The string in UpperCase: " + st.toUpperCase());

        System.out.println("The Substring from the inputted string is " + st.substring(4,11));

        char oldchar = 'a';
        char newchar = 'e';

        String newstring = st.replace(oldchar,newchar);

        System.out.println("The New string is " + newstring);
    }
}
```

Output

```
Enter a String
The kindom has fallen
The length of the string is 21
The string in UpperCase: THE KINDOM HAS FALLEN
The Substring from the inputted string is kindom
The New string is The kindom hes fellen
```

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:

```java
public class Parse {

    public static void main(String[] args) {

        String str = "25000";
        System.out.println("The given string is : " + str);

        int i = Integer.parseInt(str);
        System.out.println("Converting string to integer : " + i);

        double d = Double.parseDouble(str);
        System.out.println("Converting string to double : " + d);

        float f = Float.parseFloat(str);
        System.out.println("Converting string to float :  " + f);

        String s = "true";
        System.out.println("The string : " + s);
        boolean b = Boolean.parseBoolean(s);
        System.out.println("Converting string to boolean : " + b);

        String n = String.valueOf(i);
        System.out.println("Converting int " + i + " back to string : " + n);

        String m = String.valueOf(d);
        System.out.println("Converting double " + d + " back to string : " + m);

        String x = String.valueOf(f);
        System.out.println("Converting float " + f + " back to string : " + x);

        String y = String.valueOf(b);
        System.out.println("Converting boolean " + b + " back to string : " + y);
    }
}
```

Output:

```
The given string is : 25000
Converting string to integer : 25000
Converting string to double : 25000.0
Converting string to float :  25000.0
The string : true
Converting string to boolean : true
Converting int 25000 back to string : 25000
Converting double 25000.0 back to string : 25000.0
Converting float 25000.0 back to string : 25000.0
Converting boolean true back to string : true
```

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

Code:

```java
public class BubbleSort {
    public static void main(String[] args) {
        int arr[] = {24, 4, 67, -2, 67, 3, 100, 90, 87};

        System.out.println("Original array:");
        for (int k : arr) {
            System.out.print(k + " ");
        }
        System.out.println();

        int temp = 0;
        for (int i = 0; i < arr.length; i++) {
            for (int j = 1; j < arr.length - i ; j++) {
                if (arr[j-1] > arr[j]){
                    temp = arr[j-1];
                    arr[j-1] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        System.out.println("Sorted array:");
        for (int j : arr) {
            System.out.print(j + " ");
        }
    }
}
```

Output:

```
Original array:
24 4 67 -2 67 3 100 90 87
Sorted array:
-2 3 4 24 67 67 87 90 100
```

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 {
    String title;
    String author;
    String isbn;
    int price;

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



    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " + isbn + ", Price: $" + price;
```

```
    }
}

public class Library {
    ArrayList<Book> books;

    public Library() {
        books = new ArrayList<>();
    }

    public void addBook(Book book) {
        books.add(book);
        System.out.println("Book added: " + book);
    }

    public void removeBook(String isbn) {
        Book bookToRemove = null;
        int bookCount = books.size();
        for (int i = 0; i < bookCount; i++) {
            Book book = books.get(i);
            if (book.isbn.equals(isbn)) {
                bookToRemove = book;
                break;
            }
        }

        if (bookToRemove != null) {
            books.remove(bookToRemove);
            System.out.println("Book removed: " + bookToRemove);
        } else {
            System.out.println("Book with ISBN " + isbn + " not
found.");
        }
    }

    public void displayBooks() {
        int bookCount = books.size();
```

```
        if (books.isEmpty()) {
            System.out.println("No books in the library.");
        } else {
            System.out.println("Books in the library:");
            for (int i = 0; i < bookCount; i++) {
                Book book = books.get(i);
                System.out.println(book);
            }
        }
    }

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

        while (true) {
            System.out.println("\nLibrary Menu:");
            System.out.println("1. Add a new book");
            System.out.println("2. Remove a book");
            System.out.println("3. Display all books");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine();  // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter book title: ");
                    String title = sc.nextLine();
                    System.out.print("Enter book author: ");
                    String author = sc.nextLine();
                    System.out.print("Enter book ISBN: ");
                    String isbn = sc.nextLine();
                    System.out.print("Enter book price: ");
                    int price = sc.nextInt();
                    sc.nextLine();  // Consume newline
```

```
                    Book newBook = new Book(title, author, isbn,
price);
                    library.addBook(newBook);
                    break;
                case 2:
                    System.out.print("Enter ISBN of the book to
remove: ");
                    String isbnToRemove = sc.nextLine();
                    library.removeBook(isbnToRemove);
                    break;
                case 3:
                    library.displayBooks();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    sc.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please
try again.");
            }
        }
    }
}
```

Output:

```
Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Exit
Choose an option: 3
Books in the library:
Title: Jungle Book, Author: Kipling, ISBN: 1249, Price: $800
Title: Harry Potter 1, Author: Rowling, ISBN: 1236, Price: $400
```

```
Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Exit
Choose an option: 2
Enter ISBN of the book to remove: 1236
Book removed: Title: Harry Potter 1, Author: Rowling, ISBN: 1236, Price: $400

Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Exit
Choose an option: 3
Books in the library:
Title: Jungle Book, Author: Kipling, ISBN: 1249, Price: $800

Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Exit
```

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

7. Implement a custom 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.ArrayList;
import java.util.HashSet;
import java.util.Scanner;

class Book1 {
    String title;
    String author;
    String isbn;
    int price;
    String genre;
    public Book1(String title, String author, String isbn, int price, String genre) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.price = price;
        this.genre = genre;
    }
    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " + isbn + ", Price: $" + price + ", Genre: "+ genre;
    }
}


public class LibraryHashSet {
```

```
    ArrayList<Book1> books = new ArrayList<>();
    HashSet<String> genres = new HashSet<>();
    public void addBook(Book1 book) {
        books.add(book);
        genres.add(book.genre);
        System.out.println("Book added: " + book);
    }

    public void removeBook(String isbn) throws
ProductNotFoundException{
        Book1 bookToRemove = null;
        int bookCount = books.size();
        for (int i = 0; i < bookCount; i++) {
            Book1 book = books.get(i);
            if (book.isbn.equals(isbn)) {
                bookToRemove = book;
                break;
            }
        }

        if (bookToRemove != null) {
            books.remove(bookToRemove);
            System.out.println("Book removed: " + bookToRemove);
        } else {
            throw new ProductNotFoundException("Book with ISBN
"+ isbn+ " not found");
        }
    }

    public void displaygenre() {
        String[] genreList = genres.toArray(new String[0]);
        if (genres.isEmpty()){
            System.out.println("No Genres Present");
        } else {
            for (int i = 0; i < genreList.length ; i++) {
                String genre = genreList[i];
                System.out.println(genre);
```

```
            }
        }
    }

    public void displayBooks() {
        int bookCount = books.size();
        if (books.isEmpty()) {
            System.out.println("No books in the library.");
        } else {
            System.out.println("Books in the library:");
            for (int i = 0; i < bookCount; i++) {
                Book1 book = books.get(i);
                System.out.println(book);
            }
        }
    }

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

        while (true) {
            System.out.println("\nLibrary Menu:");
            System.out.println("1. Add a new book");
            System.out.println("2. Remove a book");
            System.out.println("3. Display all books");
            System.out.println("4. Display all genre");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine();  // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter book title: ");
                    String title = sc.nextLine();
                    System.out.print("Enter book author: ");
```

```
                    String author = sc.nextLine();
                    System.out.print("Enter book ISBN: ");
                    String isbn = sc.nextLine();
                    System.out.print("Enter book genre: ");
                    String genre = sc.nextLine();
                    System.out.print("Enter book price: ");
                    int price = sc.nextInt();
                    sc.nextLine();  // Consume newline
                    Book1 newBook = new Book1(title, author,
isbn, price, genre);
                    library.addBook(newBook);
                    break;
                case 2:
                    System.out.print("Enter ISBN of the book to
remove: ");
                    String isbnToRemove = sc.nextLine();
                    try {
                        library.removeBook(isbnToRemove);
                    } catch (ProductNotFoundException e){
                        System.out.println(e.getMessage());
                    }
                    break;
                case 3:
                    library.displayBooks();
                    break;
                case 4:
                    library.displaygenre();
                    break;
                case 5:
                    System.out.println("Exiting...");
                    sc.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please
try again.");
            }
        }
```

```
    }
}
```

Code:

```
Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Display all genre
5. Exit
Choose an option: 1
Enter book title: th
Enter book author: ruth
Enter book ISBN: 14522
Enter book genre: fiction
Enter book price: 1200
Book added: Title: th, Author: ruth, ISBN: 14522, Price: $1200, Genre: fiction

Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Display all genre
5. Exit
Choose an option: 1
Enter book title: pot
Enter book author: oppt
Enter book ISBN: 1236
Enter book genre: fiction
Enter book price: 100
Book added: Title: pot, Author: oppt, ISBN: 1236, Price: $100, Genre: fiction
```

```
Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Display all genre
5. Exit
Choose an option: 3
Books in the library:
Title: th, Author: ruth, ISBN: 14522, Price: $1200, Genre: fiction
Title: pot, Author: oppt, ISBN: 1236, Price: $100, Genre: fiction

Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Display all genre
5. Exit
Choose an option: 4
fiction

Library Menu:
1. Add a new book
2. Remove a book
3. Display all books
4. Display all genre
5. Exit
Choose an option: 2
Enter ISBN of the book to remove: 250
Book with ISBN 250 not found
```

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

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

class Store {
    String title;
    int isbn;

    public Store(int isbn, String title){
        this.isbn = isbn;
        this.title = title;

    }

    @Override
    public String toString() {
        return "Title: " + title + " ID: " + isbn;
    }
}


public class LibraryHashMap {

    public void addBook(Store store) {
        bookMap.put(store.isbn, store);
        System.out.println("Book Added: " + store);
    }

    public void updateBook(int isbn, String title){
        Store store = bookMap.get(isbn);
        if(store!=null){
            store.title = title;
            System.out.println("Book Updated: " + store);
        } else {
```

```
            System.out.println("Book with ISBN " + isbn + " not
found");
        }
    }

    public Store getBook(int isbn){
        return bookMap.get(isbn);
    }

    public void displayBook(){
        if(bookMap.isEmpty()){
            System.out.println("No Books in the library");
        } else {
            System.out.println("Books in the library are: \n");
            for(Store store : bookMap.values()){
                System.out.println(store);
            }
        }
    }
    HashMap<Integer, Store> bookMap= new HashMap<>();
    public static void main(String[] args) {
        LibraryHashMap library = new LibraryHashMap();
        Scanner sc = new Scanner(System.in);

        while (true){
            System.out.println("\nLibrary Menu:");
            System.out.println("1. Add a new book");
            System.out.println("2. Update a book");
            System.out.println("3. Retrieve a book");
            System.out.println("4. Display all books");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");

            int choice = sc.nextInt();
            sc.nextLine();
            switch (choice){
                case 1:
```

```
                    System.out.println("Enter the title");
                    String title = sc.nextLine();
                    System.out.println("Enter the ISBN");
                    int isbn = sc.nextInt();
                    Store newBook = new Store(isbn, title);
                    library.addBook(newBook);
                    break;
                case 2:
                    System.out.println("Enter the ISBN of book
you want to update");
                    int updateisbn = sc.nextInt();
                    sc.nextLine();
                    System.out.println("Enter the new title");
                    String updateTitle = sc.nextLine();
                    library.updateBook(updateisbn, updateTitle);
                    break;
                case 3 :
                    System.out.println("Enter the ISBN of book
you want to retrieve");
                    int retrieveisbn = sc.nextInt();
                    Store retrieveBook  =
library.getBook(retrieveisbn);
                    if (retrieveBook!=null){
                        System.out.println("Book details: " +
retrieveBook);
                    } else {
                        System.out.println("No Book found");
                    }
                    break;
                case 4:
                    library.displayBook();
                    break;
                case 5:
                    System.out.println("Exiting....");
                    sc.close();
                    break;
                default:
```

```
                    System.out.println("Invalid Choice");
                }

            }
        }
}
```

Output:

```
Library Menu:
1. Add a new book
2. Update a book
3. Retrieve a book
4. Display all books
5. Exit
Choose an option: 1
Enter the title
Jungle
Enter the ISBN
123
Book Added: Title: Jungle ID: 123

Library Menu:
1. Add a new book
2. Update a book
3. Retrieve a book
4. Display all books
5. Exit
Choose an option: 1
Enter the title
Haary Potter
Enter the ISBN
121
Book Added: Title: Haary Potter ID: 121
```

```
Library Menu:
1. Add a new book
2. Update a book
3. Retrieve a book
4. Display all books
5. Exit
Choose an option: 2
Enter the ISBN of book you want to update
121
Enter the new title
Harry Potter
Book Updated: Title: Harry Potter ID: 121

Library Menu:
1. Add a new book
2. Update a book
3. Retrieve a book
4. Display all books
5. Exit
Choose an option: 3
Enter the ISBN of book you want to retrieve
120
No Book found

Library Menu:
1. Add a new book
2. Update a book
3. Retrieve a book
4. Display all books
5. Exit
Choose an option: 4
Books in the library are:

Title: Harry Potter ID: 121
Title: Jungle ID: 123
```

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.*;

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

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

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

public class readWrite {
    public static void readEmployee(String fileName) {
        try (BufferedReader br = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = br.readLine()) != null) {
                if (line.startsWith("ID")) {
                    continue;
                }
                String[] details = line.split(",");
                int id = Integer.parseInt(details[0]);
                String name = details[1];
                String position = details[2];
```

```
                Employee employee = new Employee(id, name,
position);
                System.out.println("Read Employee: " +
employee);
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading
the file");
            e.printStackTrace();
        }
    }

    public static void writeEmployee(String fileName, Employee[]
employees) {
        try (BufferedWriter bw = new BufferedWriter(new
FileWriter(fileName, true))) {
            for (Employee employee : employees) {
                bw.write(employee.toString());
                bw.newLine();
            }
            System.out.println("Employee details written");
        } catch (IOException e) {
            System.out.println("An error occurred during
writing");
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        String inputFileName = "Java Challenges/employee.txt";
        String outputFileName = "Java Challenges/employee.txt";

        readEmployee(inputFileName);

        Employee[] newEmployees = {
                new Employee(4, "Virat", "Director"),
                new Employee(5, "Dhoni", "Cleaner")
```

```
        };

        writeEmployee(outputFileName, newEmployees);
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-20\bin\java.
Read Employee: 1,Siv hari,Developer
Read Employee: 2,Vivek,Manager
Read Employee: 3,abhinav,Analyst
Employee details written
```

```
ID,Name,Position
1,Siv hari,Developer
2,Vivek,Manager
3,abhinav,Analyst
```

```
ID,Name,Position
1,Siv hari,Developer
2,Vivek,Manager
3,abhinav,Analyst
4,Virat,Director
5,Dhoni,Cleaner
```