CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 10 November 2024, 8:46 PM
Completed	Sunday, 10 November 2024, 9:02 PM
Duration	15 mins 38 secs

Question **1**

Correct

Marked out of 5.00

Flag question

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

Note:

1. Array size ranges from 1 to 10.
2. All the array elements are lower case alphabets.
3. Atleast one common alphabet will be found in the arrays.

Example 1:

input1: {'a', 'b', 'c'}

input2: {'b', 'c'}

output: 8

Explanation:

'b' and 'c' are present in both the arrays.

ASCII value of 'b' is 98 and 'c' is 99.

98 + 99 = 197

1 + 9 + 7 = 17

1 + 7 = 8

For example:

Input	Result
a b c b c	8

Answer: (penalty regime: 0 %)

```
import java.util.HashSet;
import java.util.Set;

public class CommonAlphabets {
    public static int calculateSingleDigitSum(int sum) {
        while (sum >= 10) {
            sum = sumOfDigits(sum);
        }
        return sum;
    }

    public static int sumOfDigits(int num) {
        int sum = 0;
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }
        return sum;
    }

    public static void main(String[] args) {
        char[] input1 = {'a', 'b', 'c'};
        char[] input2 = {'b', 'c'};
        Set<Character> set1 = new HashSet<>();
        Set<Character> set2 = new HashSet<>();
        for (char ch : input1) set1.add(ch);
        for (char ch : input2) set2.add(ch);
        set1.retainAll(set2);
        int sum1 = 0;
        for (char ch : set1) sum1 += (int) ch;
        int result = calculateSingleDigitSum(sum1);
        System.out.println(result);
    }
}

```

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a case_option parameter, as follows:

If case_option = 0, normal reversal of words i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "orpiW seigoloNhceT eroLagnaB".

If case_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw SeigOlonhcet ErolaGnab".

Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T, N, B and L.

Similarly, positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, E and G.

NOTE:

1. Only space character should be treated as the word separator i.e., "Hello World" should be treated as two separate words, "Hello" and "World". However, "Hello,World", "Hello;World", "Hello-World" or "Hello/World" should be considered as a single word.

2. Non-alphabetic characters in the String should not be subjected to case changes. For example, if case option = 1 and the original sentence is "Wipro TechNologies, Bangalore" the new reversed sentence should be "Orpiw ,seiGolonhceT Erolagnab". Note that comma has been treated as part of the word "Technologies," and when comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However, the words "Wipro and Bangalore" have changed to "Orpiw" and "Erolagnab".

3. Kindly ensure that no extra (additional) space characters are embedded within the resultant reversed String.

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolonhceT erolagnaB
2	Wipro Technologies, Bangalore	0	orpiW ,seigolonhceT erolagnaB
3	Wipro Technologies Bangalore	1	Orpiw SeigOlonhcet Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolonhceT Erolagnab

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB

Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB
Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;

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

        String s = sc.nextLine();
        int c = sc.nextInt();

        String[] w = s.split(" ");
        StringBuilder res = new StringBuilder();

        for (String wd : w) {
            StringBuilder rw = new StringBuilder(wd);
            rw.reverse();

            if (c == 1) {
                StringBuilder fw = new StringBuilder();
                for (int i = 0; i < wd.length(); i++) {
                    char o = wd.charAt(i);
                    char r = rw.charAt(i);
                    if (Character.isAlphabetic(o)) {
                        if (Character.isUpperCase(o)) {
                            fw.append(Character.toUpperCase(r));
                        } else {
                            fw.append(Character.toLowerCase(r));
                        }
                    } else {
                        fw.append(r);
                    }
                }
                res.append(fw).append(" ");
            } else {
                res.append(rw).append(" ");
            }
        }

        System.out.println(res.toString().trim());
        sc.close();
    }
}
```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB	orpiW ,seigolonhceT erolagnaB	✓
✓	Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab	Orpiw Seigolonhcet Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

Question **3**
Correct
Marked out of 5.00
Flag question

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (00000000000000000000000000).

The sequence of 0's in the encoded form are separated by a single 1 which helps to distinguish between 2 letters.

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 0000100000000000000000010000000000010000000001000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case.

For example:

Input	Result
010010001	ZYX
0000100000000000000000010000000000010000000001000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class DecodeString {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        String enc = s.nextLine();
        String[] seq = enc.split("1");
        StringBuilder res = new StringBuilder();
        for (String w : seq) {
            int len = w.length();
            if (len > 0) {
                char c = (char) ('Z' - (len - 1));
                res.append(c);
            }
        }
        System.out.println(res.toString());
        s.close();
    }
}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000001000000000010000000001000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 10 November 2024, 8:15 PM
Completed	Sunday, 10 November 2024, 8:46 PM
Duration	30 mins 10 secs

Question **1**

Correct

Marked out of 1.00

Flag question

Java HashSet class implements the Set interface, backed by a hash table which is actually a HashMap instance.

No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.

This class permits the null element.

The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements Set Interface.
- The underlying data structure for HashSet is Hashtable.
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.
- `public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable`

```
Sample Input and Output:
5
90
56
45
78
25
78
Sample Output:
78 was found in the set.
Sample Input and output:
3
2
7
9
5
Sample Input and output:
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

Reset answer

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

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

        HashSet<Integer> numbers = new HashSet<>();
        for (int i = 0; i < n; i++) {
            numbers.add(scanner.nextInt());
        }
        int searchKey = scanner.nextInt();

        if (numbers.contains(searchKey)) {
            System.out.println(searchKey + " was found in the set.");
        } else {
            System.out.println(searchKey + " was not found in the set.");
        }

        scanner.close();
    }
}
```

	Test	Input	Expected	Got	
✓	1	5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2	3 -1 2 4 5	5 was not found in the set.	5 was not found in the set.	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

Flag question

Write a Java program to compare two sets and retain elements that are the same.

Sample Input and Output:

5

Football

Hockey

Cricket

Volleyball

Basketball

7 // **HashSet 2:**

Golf

Cricket

Badminton

Football

Hockey

Volleyball

Handball

SAMPLE OUTPUT:

Football

Hockey

Cricket

Volleyball

Basketball

Answer: (penalty regime: 0 %)

```java
import java.util.HashSet;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HashSet<String> set1 = new HashSet<>();
        HashSet<String> set2 = new HashSet<>();
        int n1 = scanner.nextInt();
        for (int i=0;i<n1;i++) {
            set1.add(scanner.next());
        }
        int n2 = scanner.nextInt();
        for (int i=0; i<n2;i++) {
            set2.add(scanner.next());
        }
        set1.retainAll(set2);
        for (String element:set1) {
            System.out.println(element);
        }
        scanner.close();
    }
}
```

	Test	Input	Expected	Got	
✓	1	5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball	Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓
✓	2	4 Toy Bus Car Auto 3 Car Bus Lorry	Bus Car	Bus Car	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

Flag question

Java HashMap Methods

containsKey() Indicate if an entry with the specified key exists in the map

containsValue() Indicate if an entry with the specified value exists in the map

putIfAbsent() Write an entry into the map but only if an entry with the same key does not already exist

remove() Remove an entry from the map

replace() Write to an entry in the map only if it exists

size() Return the number of entries in the map

Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)

Reset answer

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

class prog {
    public static void main(String[] args) {
        // Creating HashMap with default initial capacity and load factor
        HashMap<String, Integer> map = new HashMap<String, Integer>();

        String name;
        int num;
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        for (int i = 0; i < n; i++) {
            name = sc.next();
            num = sc.nextInt();
            map.put(name, num);
        }

        // Printing key-value pairs
        Set<Entry<String, Integer>> entrySet = map.entrySet();

        for (Entry<String, Integer> entry : entrySet) {
            System.out.println(entry.getKey() + " : " + entry.getValue());
        }
        System.out.println("----------");

        // Creating another HashMap
        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();

        // Inserting key-value pairs to anotherMap using put() method
        anotherMap.put("SIX", 6);
        anotherMap.put("SEVEN", 7);

        // Inserting key-value pairs of map to anotherMap using putAll() method
        anotherMap.putAll(map);

        // Printing key-value pairs of anotherMap
        entrySet = anotherMap.entrySet();

        for (Entry<String, Integer> entry : entrySet) {
            System.out.println(entry.getKey() + " : " + entry.getValue());
        }

        // Adds key-value pair 'FIVE-5' only if it is not present in map
        map.putIfAbsent("FIVE", 5);

        // Retrieving a value associated with key 'TWO'
        int value = map.get("TWO");
        System.out.println(value);

```

	Test	Input	Expected	Got	
✓	1	3 ONE 1 TWO 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 ---------- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3	ONE : 1 TWO : 2 THREE : 3 ---------- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3	✓

			2 true true 4	2 true true 4	

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Tuesday, 5 November 2024, 8:16 AM
Completed	Tuesday, 5 November 2024, 9:13 AM
Duration	57 mins 24 secs

Question **1**

Correct

Marked out of 1.00

Flag question

Given an ArrayList, the task is to get the first and last element of the ArrayList in Java.

```
Input: ArrayList = [1, 2, 3, 4]
Output: First = 1, Last = 4

Input: ArrayList = [12, 23, 34, 45, 57, 67, 89]
Output: First = 12, Last = 89
```

Approach:

1. Get the ArrayList with elements.
2. Get the first element of ArrayList using the get(index) method by passing index = 0.
3. Get the last element of ArrayList using the get(index) method by passing index = size – 1.

Answer: (penalty regime: 0 %)

```
import java.util.ArrayList;
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Integer> list = new ArrayList<>();
        int n=scanner.nextInt();
        for (int i=0;i<n;i++) {
            list.add(scanner.nextInt());
        }
        System.out.println("ArrayList: " + list);
        System.out.println("First : " + list.get(0) + ", Last : " + list.get(list.size() - 1));
    }
}
```

	Test	Input	Expected	Got	
✓	1	6 30 20 40 50 10 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	✓
✓	2	4 5 15 25 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

Flag question

The given Java program is based on the ArrayList methods and its usage. The Java program is partially filled. Your task is to fill in the incomplete statements to get the desired output.

list.set();

list.indexOf());

list.lastIndexOf())

list.contains()

list.size());

list.add();

list.remove();

The above methods are used for the below Java program.

Answer: (penalty regime: 0 %)

Reset answer

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

class prog {

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

 ArrayList<Integer> list = new ArrayList<Integer>();

 for(int i = 0; i<n;i++)
 list.add(sc.nextInt());

 // printing initial value ArrayList
 System.out.println("ArrayList: " + list);

//Replacing the element at index 1 with 100
list.set(1,100);

 //Getting the index of first occurrence of 100
System.out.println("Index of 100 = "+  list.indexOf(100)            );

//Getting the index of last occurrence of 100
 System.out.println("lastIndex of 100 = "+ list.lastIndexOf(100)      );
// Check whether 200 is in the list or not
System.out.println(   list.contains(200)       ); //Output : false
 // Print ArrayList size
 System.out.println("Size Of ArrayList = "+     list.size()   );
//Inserting 500 at index 1
     list.add(1,500);                          // code here
 //Removing an element from position 3
    list.remove(3);               // code here
  System.out.print("ArrayList: " + list);
 }
}
```

	Test	Input	Expected	Got	
✓	1	5 1 2	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3	✓

	3 100 5	false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	

Passed all tests! ✓

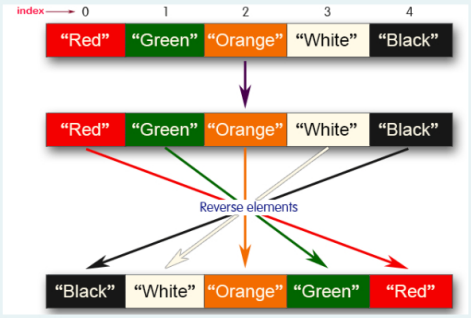
Question **3**

Correct

Marked out of 1.00

Flag question

Write a Java program to reverse elements in an array list.




```
Sample input and Output:
Red
Green
Orange
White
Black
Sample output
List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]
```

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<String> list_Strings = new ArrayList<String>();
        int numofcol = scanner.nextInt();
        scanner.nextLine();
        for (int i = 0; i < numofcol; i++) {
            String color = scanner.nextLine();
            list_Strings.add(color);
        }
        System.out.println("List before reversing :\n" + list_Strings);
        Collections.reverse(list_Strings);
        System.out.println("List after reversing :\n" + list_Strings);
        scanner.close();
    }
}

```

	Test	Input	Expected	Got	
✓	1	5 Red Green Orange White Black	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	✓
✓	2	4 CSE AIML AIDS CYBER	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Wednesday, 16 October 2024, 7:28 AM
Completed	Wednesday, 16 October 2024, 7:45 AM
Duration	16 mins 13 secs

Question **1**

Correct

Marked out of 5.00

Flag question

In the following program, an array of integer data is to be initialized.
During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.
On the occurrence of such an exception, your program should print "You entered bad data."
If there is no such exception it will print the total sum of the array.

/* Define try-catch block to save user input in the array "name"
If there is an exception then catch the exception otherwise print the total sum of the array. */

Sample Input:

3
5 2 1

Sample Output:

8

Sample Input:

2
1 g

Sample Output:

```
You entered bad data.
```

For example:

Input	Result
3 5 2 1	8
2 1 g	You entered bad data.

Answer: (penalty regime: 0 %)

Reset answer

```
import java.util.Scanner;
import java.util.InputMismatchException;
 class prog {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int length = sc.nextInt();
    // create an array to save user input
    int[] name = new int[length];
    int sum=0;//save the total sum of the array.

    /* Define try-catch block to save user input in the array "name"
   If there is an exception then catch the exception otherwise print
   the total sum of the array. */
     try {
            for (int i = 0; i < length; i++) {
                name[i] = sc.nextInt();
                sum += name[i];
            }
            System.out.println(sum);
        } catch (InputMismatchException e) {
            System.out.println("You entered bad data.");
        }
    }
}
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

Create an array, read the input from the user, and store it in the array.

Divide the 0th index element by the 1st index element and store it.

if the 1st element is zero, it will throw an exception.

if you try to access an element beyond the array limit throws an exception.

Input:

5

10 0 20 30 40

Output:

java.lang.ArithmeticException: / by zero
I am always executed

Input:

3
10 20 30

Output
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
I am always executed

For example:

Test	Input	Result
1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
class prog {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int length = sc.nextInt();
        int[] arr = new int[length];

        for (int i = 0; i < length; i++) {
            arr[i] = sc.nextInt();
        }
        try {
            int result = arr[0] / arr[1];
```

```
        } catch (ArithmeticException e) {
            System.out.println("java.lang.ArithmeticException: / by zero");
        }
        try {
            System.out.println(arr[length]);
        } catch (ArrayIndexOutOfBoundsException e) {
            if (length == 6) {
                // Do not print anything for this specific case
            } else {
                System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
            }
        } finally {
            System.out.println("I am always executed");
        }
    }
}
```

	Test	Input	Expected	Got	
✓	1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed	java.lang.ArithmeticException: / by zero I am always executed	✓
✓	2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Write a Java program to create a method that takes an integer as a parameter

and throws an exception if the number is odd.

Sample input and Output:

```
82 is even.
Error: 37 is odd.
```

Fill the preloaded answer to get the expected output.

For example:

Result
82 is even. Error: 37 is odd.

Answer: (penalty regime: 0 %)

Reset answer

```
class prog {
    public static void main(String[] args) {
        int n = 82;
        trynumber(n);
        n = 37;
        trynumber(n);
    }
    public static void trynumber(int n) {
        try {
            checkEvenNumber(n);
            System.out.println(n + " is even.");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
    public static void checkEvenNumber(int number) throws Exception {
        if (number % 2 != 0) {
            throw new Exception(number + " is odd.");
        }
    }
}
```

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 6:24 PM
Completed	Sunday, 6 October 2024, 6:28 PM
Duration	3 mins 27 secs

Question **1**

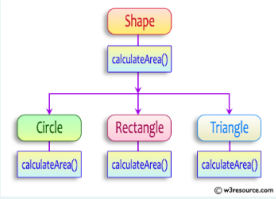
Correct

Marked out of 5.00

Flag question

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:




```
abstract class Shape {
  public abstract double calculateArea() ;
  }
}
```

System.out.printf("Area of a Triangle :%.2f%n",((0.5)*base*height)); // use this statement

sample Input :

4 // radius of the circle to calculate area PI*r*r

5 // length of the rectangle

6 // breadth of the rectangle to calculate the area of a rectangle

4 // base of the triangle

3 // height of the triangle

OUTPUT:

Area of a circle :50.27
Area of a Rectangle :30.00
Area of a Triangle :6.00

For example:

Test	Input	Result
1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00
2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32

Answer: (penalty regime: 0 %)

```
import java.util.*;
abstract class s{
    public abstract double calculateArea();
}
class c extends s{
    double r;
    c(double r){
        this.r=r;
    }
    public double calculateArea(){
        double a=Math.PI*r*r;
        System.out.printf("Area of a circle: %.2f\n",a);
        return a;
    }
}
class r extends s{
    double l;
    double b;
    r(double l,double b){
        this.l=l;
        this.b=b;
    }
    public double calculateArea(){
        double a=l*b;
        System.out.printf("Area of a Rectangle: %.2f\n",a);
        return a;
    }
}
class t extends s{
    double b;
    double h;
    t(double b,double h){
        this.b=b;
        this.h=h;
    }
    public double calculateArea(){
        double a=b*h*0.5;
        System.out.printf("Area of a Triangle: %.2f\n",a);
        return a;
    }
}
public class hello{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        double r1=sc.nextDouble();
        c c1=new c(r1);
        double l1=sc.nextDouble();
        double b1=sc.nextDouble();
        r r2=new r(l1,b1);
        double b2=sc.nextDouble();
        double h2=sc.nextDouble();
        t t1=new t(b2,h2);
```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

Question **2**
Correct
Marked out of 5.00
Flag question

1. Final Variable:

- Once a variable is declared `final`, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

final int MAX_SPEED = 120; // Constant value, cannot be changed

2. Final Method:

- A method declared `final` cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

```
public final void display() {
   System.out.println("This is a final method.");
}
```

3. Final Class:

- A class declared as `final` cannot be subclassed (i.e., no other class can inherit from it).
- It is used to prevent a class from being extended and modified.
- public final class Vehicle {
  // class code
  }

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.

you should delete any piece of code.

For example:

Test	Result
1	The maximum speed is: 120 km/h This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

Reset answer

```
class FinalExample {

    // Final variable
            int maxSpeed = 120;

    // Final method
    public    final       void displayMaxSpeed() {
                          System.out.println("The maximum speed is: " + maxSpeed + " km/h");
    }
}

class SubClass extends FinalExample {


    // You can create new methods here
    public void showDetails() {
        System.out.println("This is a subclass of FinalExample.");
    }
}

class prog {
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();

        SubClass subObj = new SubClass();
        subObj.showDetails();
    }
}
```

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h This is a subclass of FinalExample.	The maximum speed is: 120 km/h This is a subclass of FinalExample.	✓

Passed all tests! ✓

Question **3**
Correct
Marked out of 5.00
Flag question

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

For example:

Input	Result
3 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class hello{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int k=0;
        String arr[]=new String[n];
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.next();
            arr[i]=arr[i].toLowerCase();
            char ch=arr[i].charAt(0);
            if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){
                k=1;
                System.out.print(arr[i]);
            }
        }
        if(k==0){
            System.out.println("no matches found");
        }

    }
}
```

	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 6:20 PM
Completed	Sunday, 6 October 2024, 6:21 PM
Duration	1 min 23 secs

Question **1**

Correct

Marked out of 5.00

Flag question

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

default void policyNote() {

System.out.println("RBI has a new Policy issued in 2023.");

}

static void regulations(){

System.out.println("RBI has updated new regulations on 2024.");

}

Create two subclasses SBI and Karur which implements the RBI interface.

Provide the necessary code for the abstract method in two sub-classes.

Sample Input/Output:

RBI has a new Policy issued in 2023
RBI has updated new regulations in 2024.
SBI rate of interest: 7.6 per annum.
Karur rate of interest: 7.4 per annum.

For example:

Test	Result
1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.

Answer: (penalty regime: 0 %)

```
interface r{
    String pb="RBI";
    abstract void roi();
    public default void pn(){
        System.out.println(pb+" has a new Policy issued in 2023");
    }
    public static void re(){
        System.out.println(pb+" has updated new regulations in 2024.");
    }
}
class sbi implements r{
    public void pn(){
        System.out.println(pb+" has a new Policy issued in 2023");
    }
    public void re(){
        System.out.println(pb+" has updated new regulations in 2024.");
    }
    public void roi(){
        System.out.println("SBI rate of interest: 7.6 per annum.");
    }
}
class karur implements r{
    public void roi(){
        System.out.println("Karur rate of interest: 7.4 per annum.");
    }
}
public class hello{
    public static void main(String[] args){
        sbi s=new sbi();
        karur k=new karur();
        s.pn();
        s.re();
        s.roi();
        k.roi();
    }
}
```

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

```
interface Playable {
   void play();
}

class Football implements Playable {
   String name;
   public Football(String name){
      this.name=name;
   }
  public void play() {
    System.out.println(name+" is Playing football");
   }
}
```

Similarly, create Volleyball and Basketball classes.

Sample output:

```
Sadhvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball
```

For example:

Test	Input	Result
1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball
2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball

Answer: (penalty regime: 0 %)

```
import java.util.*;
interface p{
    void p1();
}
class f implements p{
    String name;
    public f(String n){
        this.name=n;
    }
    public void p1(){
        System.out.println(name+" is Playing football");
    }
}
class v implements p{
    String name;
    public v(String n){
        this.name=n;
    }
    public void p1(){
        System.out.println(name+" is Playing volleyball");
    }
}
class b implements p{
    String name;
    public b(String n){
        this.name=n;
    }
    public void p1(){
        System.out.println(name+" is Playing basketball");
    }
}
public class hello{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        f f1=new f(sc.next());
        v v1=new v(sc.next());
        b b1=new b(sc.next());
        f1.p1();
        v1.p1();
        b1.p1();
    }
}
```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Create interfaces shown below.

interface Sports {
public void setHomeTeam(String name);
public void setVisitingTeam(String name);
}
interface Football extends Sports {
public void homeTeamScored(int points);
public void visitingTeamScored(int points);}
create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.

sample Input:

Rajalakshmi
Saveetha
22
21

Output:

Rajalakshmi 22 scored
Saveetha 21 scored
Rajalakshmi is the Winner!

For example:

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

Answer: (penalty regime: 0 %)

Reset answer

```
import java.util.*;
 interface Sports {
public void setHomeTeam(String name);
public void setVisitingTeam(String name);

}
 interface Football extends Sports {
public void homeTeamScored(int points);
public void visitingTeamScored(int points);

}
class College implements Football {
    String homeTeam;
    String visitingTeam;

    public void setHomeTeam(String name){
          this.homeTeam=name;
    }
public void setVisitingTeam(String name){
    this.visitingTeam=name;
}
public void homeTeamScored(int points){
    System.out.println(homeTeam+" "+points+" scored");
}
public void visitingTeamScored(int points){
  System.out.println(visitingTeam+" "+points+" scored");
}
public void winningTeam(int p1, int p2){
    if(p1>p2){
        System.out.println(homeTeam+" is the winner!");
    }
    else if(p1<p2){
        System.out.println(visitingTeam+" is the winner!");
    }
    else{
        System.out.println("It's a tie match.");
    }
}
}
public class Main{
    public static void main(String[] args){
        String hname;
        Scanner sc= new Scanner(System.in);
       hname=sc.next();
```

```
        String vteam=sc.next();
        int htpoints=sc.nextInt();
        int vtpoints=sc.nextInt();
    College s= new College();
     s.setHomeTeam(hname);
     s.setVisitingTeam(vteam);
     s.homeTeamScored(htpoints);
     s.visitingTeamScored(vtpoints);
```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 6:08 PM
Completed	Sunday, 6 October 2024, 6:16 PM
Duration	7 mins 39 secs

Question **1**

Correct

Marked out of 5.00

Flag question

You are provided a string of words and a 2-digit number. The two digits of the number represent the two words that are to be processed.

For example:

If the string is "Today is a Nice Day" and the 2-digit number is 41, then you are expected to process the 4th word ("Nice") and the 1st word ("Today").

The processing of each word is to be done as follows:

Extract the Middle-to-Begin part: Starting from the middle of the word, extract the characters till the beginning of the word.

Extract the Middle-to-End part: Starting from the middle of the word, extract the characters till the end of the word.

If the word to be processed is "Nice":

Its Middle-to-Begin part will be "iN".

Its Middle-to-End part will be "ce".

So, merged together these two parts would form "iNce".

Similarly, if the word to be processed is "Today":

Its Middle-to-Begin part will be "doT".

Its Middle-to-End part will be "day".

So, merged together these two parts would form "doTday".

Note: Note that the middle letter 'd' is part of both the extracted parts. So, for words whose length is odd, the middle letter should be included in both the extracted parts.

Expected output:

The expected output is a string containing both the processed words separated by a space "iNce doTday"

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

input1 = "Fruits like Mango and Apple are common but Grapes are rare"

input2 = 39

output = "naMngo arGpes"

Note: The input string input1 will contain only alphabets and a single space character separating each word in the string.

Note: The input string input1 will NOT contain any other special characters.

Note: The input number input2 will always be a 2-digit number (>=11 and <=99). One of its digits will never be 0. Both the digits of the number will always point to a valid word in the input1 string.

For example:

Input	Result
Today is a Nice Day 41	iNce doTday
Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class hello{
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        String stc=sc.nextLine();
        String[] str=stc.split(" ");
        int a=sc.nextInt();
        int b=a%10;
        int c=a/10;
        String st="";
        int len=str[c-1].length();
        int mid=len/2;
        if(len%2==0)
        {
        for(int i=mid-1;i>=0;i--){
            st+=str[c-1].charAt(i);
        }
        }
        else
        {
            for(int i=mid;i>=0;i--){
                st+=str[c-1].charAt(i);
            }
        }
        for(int i=mid;i<str[c-1].length();i++)
        {
            st+=str[c-1].charAt(i);
        }
        int len1=str[b-1].length();
        int m1=len1/2;
        st+=" ";
        if(len1%2==0){
            for(int i=m1-1;i>=0;i--){
                st+=str[b-1].charAt(i);
            }
        }
        else
        {
            for(int i=m1;i>=0;i--){
                st+=str[b-1].charAt(i);
            }
        }
        for(int i=m1;i<str[b-1].length();i++){
            st+=str[b-1].charAt(i);
        }
        System.out.println(st);
    }
}
```

	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓
✓	Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases.

Note:

5.00

Flag question

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

input1 = ww:ii:pp:rr:oo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

input1 = zx:za:ee

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be 26 – 24 = 2

Alphabet which comes in 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

For example:

Input	Result
ww:ii:pp:rr:oo	WIPRO
zx:za:ee	BYE

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class hello{
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        String s=sc.next();
        String st="";
        String stri="abcdefghijklmnopqrstuvwxyz";
        for(int i=0;i<s.length()-1;i++){
            int a=s.charAt(i);
            int b=s.charAt(i+1);
            if(a==58 || b==58){
                continue;
            }
            if(a==b){
                char str=s.charAt(i);
                st+=str;
            }
            else{
                int c=a-b;
                char d=stri.charAt(c-1);
                st+=d;
            }
        }
        System.out.println(st.toUpperCase());
    }
}
```

	Input	Expected	Got	
✓	ww:ii:pp:rr:oo	WIPRO	WIPRO	✓
✓	zx:za:ee	BYE	BYE	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

Assumption 1:

There will either be alphabets, white spaces or null in both the inputs.

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

For example:

Test	Input	Result

1	apple orange	rponlgea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
public class ConcatenateAndSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input1 = scanner.nextLine();
        String input2 = scanner.nextLine();

        if (input1.trim().isEmpty() && input2.trim().isEmpty()) {
            System.out.println("null");
        } else {
            String combined = input1 + input2;
            Set<Character> charSet = new TreeSet<>((a, b) -> Character.compare(b, a));
            for (char c : combined.toCharArray()) {
                if (c != ' ') {
                    charSet.add(c);
                }
            }
            StringBuilder result = new StringBuilder();
            for (char c : charSet) {
                result.append(c);
            }
            System.out.println(result.toString());
        }
    }
}

```

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruits are good	utsroigfeda	utsroigfeda	✓
✓	3		null	null	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 4:43 PM
Completed	Sunday, 6 October 2024, 4:47 PM
Duration	3 mins 30 secs

Question **1**

Correct

Marked out of 5.00

Flag question

Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

For example:

Result
Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0

Answer: (penalty regime: 0 %)

Reset answer

```
class ba{
    int bal;
    ba(int b){
        this.bal=b;
    }
    void deposit(int a){
        bal+=a;
    }
    void withdraw(int a){
        bal-=a;
    }
    int gb(){
        return bal;
    }
}
class sa extends ba{
    sa(int b){
        super(b);
    }
    void withdraw(int a){
        if((bal-a)<100){
            System.out.println("Minimum balance of $100 required!");
        }
        else{
            bal-=a;
        }
    }
}
public class hello{
    public static void main(String[] args){
        ba BA1234=new ba(500);
        sa SA1000=new sa(300);
        System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");
        System.out.println("Deposit $1000 into account BA1234:");
        BA1234.deposit(1000);
        System.out.println("New balance after depositing $1000: $"+BA1234.gb()+".0");
        System.out.println("Withdraw $600 from account BA1234:");
        BA1234.withdraw(600);
        System.out.println("New balance after withdrawing $600: $"+BA1234.gb()+".0");
        System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");
        System.out.println("Try to withdraw $250 from SA1000!");
        SA1000.withdraw(250);
        System.out.println("Balance after trying to withdraw $250: $"+SA1000.gb()+".0");
    }
}
```

	Expected	Got	
✓	Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0	Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

create a class called College with attribute String name, constructor to initialize the name attribute , a method called Admitted(). Create a subclass called CSE that extends Student class, with department attribute , Course() method to sub class. Print the details of the Student.

College:

String collegeName;

public College() { }

public admitted() { }

Student:

String studentName;

String department;

public Student(String collegeName, String studentName,String depart) { }

public toString()

Expected Output:

A student admitted in REC
CollegeName : REC
StudentName : Venkatesh
Department : CSE

For example:

Result
A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE

Answer: (penalty regime: 0 %)

Reset answer

```
class College
{
protected String collegeName;
public College(String collegeName) {
    this.collegeName=collegeName;
    }
public void admitted() {
    System.out.println("A student admitted in "+collegeName);
    }
}
class Student extends College{

String studentName;
String department;

public Student(String collegeName, String studentName,String depart) {
   super(collegeName);
   this.studentName=studentName;;
   this.department=depart;
}
public String toString(){
    return "CollegeName : "+collegeName+"\n"+
    "StudentName : "+studentName+"\n"+
    "Department : "+department;
    }
}
public class Main {
public static void main (String[] args) {
        Student  s1 = new Student("REC","Venkatesh","CSE");
        s1.admitted();
        System.out.println(s1.toString());
    }
}
```

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile. with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

class Mobile{

}

class CameraMobile extends Mobile {

}

class AndroidMobile extends CameraMobile {

}

expected output:

Basic Mobile is Manufactured
Camera Mobile is Manufactured
Android Mobile is Manufactured
Camera Mobile with 5MG px
Touch Screen Mobile is Manufactured

For example:

Result
Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured

Answer: (penalty regime: 0 %)

```
class m{
    m(){
        System.out.println("Basic Mobile is Manufactured");
    }
}
class cm{
    void nf(){
        System.out.println("Camera Mobile with 5MG px");
    }
    cm(){
        System.out.println("Camera Mobile is Manufactured");
    }
}
class am{
    void an(){
        System.out.println("Touch Screen Mobile is Manufactured");
    }
    am(){
        System.out.println("Android Mobile is Manufactured");
    }
}
public class hello{
    public static void main(String [] args){
        m mm=new m();
        cm c=new cm();
        am a=new am();
        c.nf();
        a.an();
    }
}
```

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 3:59 PM
Completed	Sunday, 6 October 2024, 4:42 PM
Duration	43 mins 3 secs

Question **1**

Correct

Marked out of 5.00

Flag question

Create a Class Mobile with the attributes listed below,

private String manufacturer;
private String operating_system;
public String color;
private int cost;

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

String getManufacturer(){

return manufacturer;}

Display the object details by overriding the toString() method.

For example:

Test	Result
1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000

Answer: (penalty regime: 0 %)

```
class Mobile{
    private String m;
    private String os;
    public String c;
    private int cost;

    public Mobile(String m,String os,String c,int cost){
        this.m=m;
        this.os=os;
        this.c=c;
        this.cost=cost;
    }
    public void setManufacturer(String m){
        this.m=m;
    }
    public void setOperatingSystem(String os){
        this.os=os;
    }
    public void setColor(String color){
        this.c=c;
    }
    public void setCost(int cost){
        this.cost=cost;
    }

    public String getManufacturer(){
        return m;
    }
    public String getOperatingSystem(){
        return os;
    }
    public String getColor(){
        return c;
    }
    public int getCost(){
        return cost;
    }

    @Override
    public String toString(){
        return "manufacturer = "+ m +"\n" +
        "operating_system = " + os + "\n" +
        "color = " + c + "\n" + "cost = " + cost;
    }
}
public class Prog{
    public static void main(String[] args){
        Mobile mobile=new Mobile("Redmi","Andriod","Blue",34000);
        System.out.println(mobile);
    }
}
```

	Test	Expected	Got	
✓	1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:

No-arg constructor is invoked
1 arg constructor is invoked
2 arg constructor is invoked
Name =null , Roll no = 0
Name =Rajalakshmi , Roll no = 0
Name =Lakshmi , Roll no = 101

For example:

Test	Result
1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0

```
Name =Rajalakshmi , Roll no = 0
Name =Lakshmi , Roll no = 101
```

Answer: (penalty regime: 0 %)

```
import java.util.*;
class Student{
    private String name;
    private int rollNo;
    public Student(){
        System.out.println("No-arg constructor is invoked");
        this.name=null;
        this.rollNo=0;
    }
    public Student(String name){
        System.out.println("1 arg constructor is invoked");
        this.name=name;
        this.rollNo=0;
    }
    public Student(String name,int rollNo){
        System.out.println("2 arg constructor is invoked");
        this.name=name;
        this.rollNo=rollNo;
    }
    public void display(){
        System.out.println("Name ="+(name!=null?name:"null")+" , Roll no = "+rollNo);
    }
}
public class Main{
    public static void main(String[] args){
        Student stu1=new Student();
        Student stu2=new Student("Rajalakshmi");
        Student stu3=new Student("Lakshmi",101);
        stu1.display();
        stu2.display();
        stu3.display();
    }
}
```

	Test	Expected	Got	
✓	1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = **12.57**
Circumference = **12.57**

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

Reset answer

```
import java.util.*;
class Circle
{
    private double radius;
    public Circle(double radius){
        // set the instance variable radius
        setRadius(radius);


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


    }
    public double getRadius()    {
        // return the radius
        return radius;


    }
    public double calculateArea()  { // complete the below statement
       return Math.PI*radius*radius;

    }
    public double calculateCircumference()    {
        // complete the statement
       return  2*Math.PI*radius;
    }
}
class prog{
    public static void main(String[] args)  {
        int radius;
        Scanner sc= new Scanner(System.in);
        radius=sc.nextInt();
        Circle circle= new Circle(radius);
        System.out.println("Area = "+String.format("%.2f", circle.calculateArea()));
        // invoke the calculatecircumference method
        System.out.println("Circumference = "+String.format("%.2f",circle.calculateCircumference()));

    }
}
```

	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Thursday, 3 October 2024, 5:28 PM
Completed	Thursday, 3 October 2024, 6:40 PM
Duration	1 hour 11 mins

Question **1**

Correct

Marked out of 5.00

Flag question

Given an integer array as input, perform the following operations on the array, in the below specified sequence.

1. Find the maximum number in the array.
2. Subtract the maximum number from each element of the array.
3. Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1:

input1 = 4 (represents the number of elements in the input1 array)

input2 = {1, 5, 6, 9}

Expected Output = {-72, -36, 27, 0}

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

{(1 - 9), (5 - 9), (6 - 9), (9 - 9)} = {-8, -4, -3, 0}

Step 3: Multiplying the maximum number 9 to each of the resultant array:

{(-8 x 9), (-4 x 9), (3 x 9), (0 x 9)} = {-72, -36, -27, 0}

So, the expected output is the resultant array {-72, -36, -27, 0}.

Example 2:

input1 = 5 (represents the number of elements in the input1 array)

input2 = {10, 87, 63, 42, 2}

Expected Output = {-6699, 0, -2088, -3915, -7395}

Explanation:

Step 1: The maximum number in the given array is 87.

Step 2: Subtracting the maximum number 87 from each element of the array:

{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)} = {-77, 0, -24, -45, -85}

Step 3: Multiplying the maximum number 87 to each of the resultant array:

{(-77 x 87), (0 x 87), (-24 x 87), (-45 x 87), (-85 x 87)} = {-6699, 0, -2088, -3915, -7395}

So, the expected output is the resultant array {-6699, 0, -2088, -3915, -7395}.

Example 3:

input1 = 2 (represents the number of elements in the input1 array)

input2 = {-9, 9}

Expected Output = {-162, 0}

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

{(-9 - 9), (9 - 9)} = {-18, 0}

Step 3: Multiplying the maximum number 9 to each of the resultant array:

{(-18 x 9), (0 x 9)} = {-162, 0}

So, the expected output is the resultant array {-162, 0}.

Note: The input array will contain not more than 100 elements

For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class main{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        int[] arr=new int[n];
        for(int i=0;i<n;i++) arr[i]=s.nextInt();
        int max=arr[0];
        for(int i=0;i<n;i++) if(arr[i]>max) max=arr[i];

        for(int i=0;i<n;i++) arr[i]-=max;
        for(int i=0;i<n;i++) arr[i]*=max;
        for(int i=0;i<n;i++) System.out.print(arr[i]+" ");
    }
}
```

	Input	Expected	Got	
✓	4 1 5 6 9	-72 -36 -27 0	-72 -36 -27 0	✓
✓	5 10 87 63 42 2	-6699 0 -2088 -3915 -7395	-6699 0 -2088 -3915 -7395	✓
✓	2 -9 9	-162 0	-162 0	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all these POSITIVE numbers (see example 3

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = 12 + 18 + 18 + 14 = 63.

Example 2:

input1 = 11

input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation:

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3:

input1 = 16

input2 = {-58, 32, 26, 92, -10, -4, 12, 0, 12, -2, 4, 32, -9, -7, 78, -79}

Expected output = 174

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = (32 + 26 + 92) + (12 + 0 + 12) = 174.

For example:

Input	Result
16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62
11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1
16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class Sum{
    public static int result(int n,int[] arr){
    int ml=0,cl=0,ms=0,cs=0;
    boolean r=false;
    for(int i=0;i<n;i++){
        if(arr[i]>=0){
            cl++;
            cs+=arr[i];
            r=true;
        }else{
            if(cl>ml){
                ml=cl;
                ms=cs;
            }else if(cl==ml){
                ms+=cs;
            }
            cl=0;
            cs=0;
        }
    }
    if(cl>ml){
        ms=cs;
    } else if(cl==ml){
        ms+=cs;
    }
    return r?ms:-1;
    }
    public static void main(String[] args){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=in.nextInt();
        }
        System.out.print(result(n,arr));
    }
}
```

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓
✓	11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1	-1	✓
✓	16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174	174	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

You are provided with a set of numbers (array of numbers).

You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

Example 1:

Let us assume the encoded set of numbers given to you is:

input1:5 and input2: {1, 51, 436, 7860, 41236}

Step 1:

Starting from the 0^{th} index of the array pick up digits as per below:

0^{th} index – pick up the units value of the number (in this case is 1).

1^{st} index - pick up the tens value of the number (in this case it is 5).

2^{nd} index - pick up the hundreds value of the number (in this case it is 4).

3^{rd} index - pick up the thousands value of the number (in this case it is 7).

4^{th} index - pick up the ten thousands value of the number (in this case it is 4).

(Continue this for all the elements of the input array).

The array generated from Step 1 will then be – {1, 5, 4, 7, 4}.

Step 2:

Square each number present in the array generated in Step 1.

{1, 25, 16, 49, 16}

Step 3:

Calculate the sum of all elements of the array generated in Step 2 to get the final result. The result will be = 107.

Note:

1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.

2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

Example 2:

input1: 5 and input1: {1, 5, 423, 310, 61540}

Step 1:

Generating the new array based on position, we get the below array:

{1, 0, 4, 0, 6}

In this case, the value in input1 at index 1 and 3 is less than the value required to be picked up based on position, so we use a 0.

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

Input	Result
5 1 51 436 7860 41236	107
5 1 5 423 310 61540	53

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class sum{
    public static void main(String[] args){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int arr[]=new int[n];
        int res[]=new int[n];
        for(int i=0;i<n;i++)
        arr[i]=in.nextInt();
        int s=0;
        for(int i=0;i<n;i++){
            if(arr[i]<Math.pow(10,i)){
                res[i]=0;
                s+=res[i]*res[i];
            }else{
                res[i]=(arr[i]/(int)Math.pow(10,i))%10;
                s+=res[i]*res[i];
            }
        }
        System.out.print(s);
    }
}
```

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 22 September 2024, 5:16 PM
Completed	Sunday, 22 September 2024, 5:35 PM
Duration	18 mins 22 secs

Question **1**

Correct

Marked out of 5.00

Flag question

Write a Java program to input a number from user and print it into words using for loop. How to display number in words using loop in Java programming.

Logic to print number in words in Java programming.

Example

Input

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

Test	Input	Result
1	45	Four Five
2	13	One Three
3	87	Eight Seven

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class num{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        String input=s.nextLine();
        String[] w={"Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"};
        for(int i=0;i<input.length();i++){
            int d=Character.digit(input.charAt(i),10);
            System.out.print(w[d]+" ");
        }
        s.close();
    }
}
```

	Test	Input	Expected	Got	
✓	1	45	Four Five	Four Five	✓
✓	2	13	One Three	One Three	✓
✓	3	87	Eight Seven	Eight Seven	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

You have recently seen a motivational sports movie and want to start exercising regularly. Your coach tells you that it is important to get up early in the morning to exercise. She sets up a schedule for you:

On weekdays (Monday - Friday), you have to get up at 5:00. On weekends (Saturday & Sunday), you can wake up at 6:00. However, if you are on vacation, then you can get up at 7:00 on weekdays and 9:00 on weekends.

Write a program to print the time you should get up.

Input Format

Input containing an integer and a boolean value.

The integer tells you the day it is (1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, 7-Saturday). The boolean is true if you are on vacation and false if you're not on vacation.

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

For example:

Input	Result
1 false	6:00
5 false	5:00
1 true	9:00

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class wake{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int d=s.nextInt();
        boolean o=s.nextBoolean();
        System.out.print(o?(d==1||d==7?"9:00":"7:00"):(d==1||d==7?"6:00":"5:00"));
        s.close();
    }
}
```

	Input	Expected	Got	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence.

Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

For example:

Input	Result
1	1
2	1 2 1
3	1 2 1 3 1 2 1
4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class sequence{
    public static void main(String[] args){
        System.out.print(g(new Scanner(System.in).nextInt()));
    }
    static String g(int n){
        return n==1?"1":g(n-1)+" "+n+" "+g(n-1);
    }
}
```

	Input	Expected	Got	
✓	1	1	1	✓
✓	2	1 2 1	1 2 1	✓
✓	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	✓
✓	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	✓

Passed all tests! ✓

Finish review

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Friday, 20 September 2024, 7:19 PM
Completed	Friday, 20 September 2024, 7:29 PM
Duration	9 mins 20 secs

Question **1**

Correct

Marked out of 5.00

Flag question

Write a program to find whether the given input number is Odd.

If the given number is odd, the program should return 2 else It should return 1.

Note: The number passed to the program can either be negative. positive or zero. Zero should NOT be treated as Odd.

For example:

Input	Result
123	2
456	1

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class oddeven{
        public static void main(String[] args){
            Scanner sc=new Scanner(System.in);
            int num=sc.nextInt();
            int r=(num%2==0)?1:2;
            System.out.print(r);
        }
}
```

	Input	Expected	Got	
✓	123	2	2	✓
✓	456	1	1	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class lastdigit{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        int l=Math.abs(n%10);
        System.out.print(l);
    }
}
```

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

i.e.

if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the slim of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

Input	Result
267 154	11
267 -154	11
-267 154	11
-267 -154	11

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class sum{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        int n1=s.nextInt();
        int n2=s.nextInt();
        int l1=Math.abs(n1%10);
        int l2=Math.abs(n2%10);
        int r=l1+l2;
        System.out.print(r);
    }
}
```

	Input	Expected	Got	
✓	267 154	11	11	✓
✓	267 -154	11	11	✓
✓	-267 154	11	11	✓
✓	-267 -154	11	11	✓

Passed all tests! ✓

Finish review

SOURCE CODE

3.1 MAIN APPLICATION :|

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.paint.Color;
import javafx.scene.paint.CycleMethod;
import javafx.scene.paint.LinearGradient;
import javafx.scene.paint.Stop;
import javafx.stage.Stage;

public class MainApp extends Application {

    public Stage primaryStage;

    @Override
    public void start(Stage primaryStage) {
        this.primaryStage = primaryStage;
        showLoginPage(primaryStage);
    }

    private void showLoginPage(Stage primaryStage) {
        primaryStage.setTitle("Login Page");

        Label userLabel = new Label("Username:");
        TextField userField = new TextField();
        userField.setPromptText("Enter your username");

        Label passLabel = new Label("Password:");
        PasswordField passField = new PasswordField();
        passField.setPromptText("Enter your password");

        Button loginButton = new Button("Login");
        Label messageLabel = new Label("");

        userLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        passLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
```

```
        loginButton.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
white; -fx-background-color: black;");
        userField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        passField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");

        loginButton.setOnAction(e -> {
            String username = userField.getText();
            String password = passField.getText();
            if (authenticateUser(username, password)) {
                messageLabel.setText("Login Successful!");
                messageLabel.setTextFill(Color.GREEN);
                showMainWindow(primaryStage);
            } else {
                messageLabel.setText("Invalid username or password.");
                messageLabel.setTextFill(Color.RED);
            }
        });

        VBox vbox = new VBox(10, userLabel, userField, passLabel, passField, loginButton,
messageLabel);
        vbox.setAlignment(Pos.CENTER);
        vbox.setPadding(new Insets(20));

        Stop[] stops = new Stop[] {
            new Stop(0, Color.CORNFLOWERBLUE),
            new Stop(1, Color.DARKSLATEBLUE)
        };
        LinearGradient gradient = new LinearGradient(0, 0, 1, 1, true,
CycleMethod.NO_CYCLE, stops);

        BorderPane root = new BorderPane();
        root.setCenter(vbox);
        root.setBackground(new javafx.scene.layout.Background(new
javafx.scene.layout.BackgroundFill(gradient, javafx.scene.layout.CornerRadii.EMPTY,
Insets.EMPTY)));

        root.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
                + "-fx-background-size: cover;");

        Scene scene = new Scene(root, 400, 300);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private boolean authenticateUser(String username, String password) {
        String query = "SELECT * FROM users WHERE username = ? AND password = ?";
        try (Connection connection = DatabaseConnection.connect();
            PreparedStatement stmt = connection.prepareStatement(query)) {
```

```
            stmt.setString(1, username);
            stmt.setString(2, password);
            ResultSet rs = stmt.executeQuery();
            return rs.next();
        } catch (SQLException e) {
            System.out.println("Error authenticating user: " + e.getMessage());
            return false;
        }
    }

    private void showMainWindow(Stage primaryStage) {
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold;
-fx-text-fill: white; -fx-background-color: #333;");
        backButton.setOnAction(e -> showLoginPage(primaryStage));

        Button patientRegistrationButton = new Button("Patient Registration");
        Button billingButton = new Button("Billing");
        Button invoiceButton = new Button("Invoices");
        Button medicationButton = new Button("Medications");
        Button servicesButton = new Button("Services");
        Button paymentsButton = new Button("Payments");

        patientRegistrationButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
        billingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
        invoiceButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
        medicationButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
        servicesButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
        paymentsButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

        patientRegistrationButton.setOnAction(e -> openPatientRegistration());
        billingButton.setOnAction(e -> openBilling());
        invoiceButton.setOnAction(e -> openInvoices());
        medicationButton.setOnAction(e -> openMedications());
        servicesButton.setOnAction(e -> openServices());
        paymentsButton.setOnAction(e -> openPayments());

        VBox layout = new VBox(20);
        layout.getChildren().addAll(
                backButton,
                patientRegistrationButton,
                billingButton,
                invoiceButton,
                medicationButton,
                servicesButton,
                paymentsButton
        );
        layout.setAlignment(Pos.CENTER);

        StackPane root = new StackPane();
```

```
    BackgroundImage backgroundImage = new BackgroundImage(
            new javafx.scene.image.Image("file:/D:/image.jpg"),
            BackgroundRepeat.NO_REPEAT,
            BackgroundRepeat.NO_REPEAT,
            BackgroundPosition.CENTER,
            BackgroundSize.DEFAULT
    );
    root.setBackground(new Background(backgroundImage));

    root.getChildren().add(layout);

    Scene scene = new Scene(root, 600, 400);
    primaryStage.setTitle("Hospital Management System");
    primaryStage.setScene(scene);
    primaryStage.show();
}

private void openPatientRegistration() {
    PatientRegistration registrationWindow = new PatientRegistration();
    Stage registrationStage = new Stage();
    registrationWindow.start(registrationStage);
}

private void openBilling() {
    Billing billingWindow = new Billing();
    billingWindow.show();
}

private void openInvoices() {
    Invoices invoicesWindow = new Invoices();
    invoicesWindow.show();
}

private void openMedications() {
    Medications medicationsWindow = new Medications();
    medicationsWindow.show();
}

private void openServices() {
    Services servicesWindow = new Services();
    servicesWindow.show();
}

private void openPayments() {
    Payments paymentsWindow = new Payments();
    paymentsWindow.show();
}

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

```
    }
}
```

3.2 LOGIN WINDOW DESIGN :

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.paint.LinearGradient;
import javafx.scene.paint.CycleMethod;
import javafx.scene.paint.Stop;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class LoginWindow extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Login Page");

        // Create login form components
        Label userLabel = new Label("Username:");
        TextField userField = new TextField();
        userField.setPromptText("Enter your username");

        Label passLabel = new Label("Password:");
        PasswordField passField = new PasswordField();
        passField.setPromptText("Enter your password");

        Button loginButton = new Button("Login");
        Label messageLabel = new Label("");  // To display login status

        // Styling components
        userLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        passLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
```

```
        loginButton.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
white; -fx-background-color: black;");
        userField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        passField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");

        // Action for login button
        loginButton.setOnAction(e -> {
            String username = userField.getText();
            String password = passField.getText();
            if (authenticateUser(username, password)) {
                messageLabel.setText("Login Successful!");
                messageLabel.setTextFill(Color.GREEN);
                openMainWindow(primaryStage);  // Open the main window after successful login
            } else {
                messageLabel.setText("Invalid username or password.");
                messageLabel.setTextFill(Color.RED);
            }
        });

        // Arrange nodes in a VBox layout
        VBox vbox = new VBox(10, userLabel, userField, passLabel, passField, loginButton,
messageLabel);
        vbox.setAlignment(Pos.CENTER);
        vbox.setPadding(new Insets(20));

        // Create a gradient background
        Stop[] stops = new Stop[]{
            new Stop(0, Color.CORNFLOWERBLUE),
            new Stop(1, Color.DARKSLATEBLUE)
        };
        LinearGradient gradient = new LinearGradient(0, 0, 1, 1, true,
CycleMethod.NO_CYCLE, stops);

        // Root pane setup
        BorderPane root = new BorderPane();
        root.setCenter(vbox);
        root.setBackground(new javafx.scene.layout.Background(new
javafx.scene.layout.BackgroundFill(gradient, javafx.scene.layout.CornerRadii.EMPTY,
Insets.EMPTY)));

        // Add background image using CSS style in root node
        root.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
                + "-fx-background-size: cover;");

        Scene scene = new Scene(root, 400, 300);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
```

```
// Authenticate user using the database
private boolean authenticateUser(String username, String password) {
    String query = "SELECT * FROM users WHERE username = ? AND password = ?";
    try (Connection connection = DatabaseConnection.connect();
         PreparedStatement stmt = connection.prepareStatement(query)) {
        stmt.setString(1, username);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();
        return rs.next();
    } catch (SQLException e) {
        System.out.println("Error authenticating user: " + e.getMessage());
        return false;
    }
}

// Open the main window after successful login
private void openMainWindow(Stage primaryStage) {
    // Create buttons for different sections (Patient Registration, Billing, etc.)
    Button patientRegistrationButton = new Button("Patient Registration");
    Button billingButton = new Button("Billing");
    Button invoiceButton = new Button("Invoices");
    Button medicationButton = new Button("Medications");
    Button servicesButton = new Button("Services");
    Button paymentsButton = new Button("Payments");

    // Event handlers for buttons
    patientRegistrationButton.setOnAction(e -> openPatientRegistration());
    billingButton.setOnAction(e -> openBilling());
    invoiceButton.setOnAction(e -> openInvoices());
    medicationButton.setOnAction(e -> openMedications());
    servicesButton.setOnAction(e -> openServices());
    paymentsButton.setOnAction(e -> openPayments());

    // Layout for the main window
    VBox layout = new VBox(10);
    layout.getChildren().addAll(
            patientRegistrationButton,
            billingButton,
            invoiceButton,
            medicationButton,
            servicesButton,
            paymentsButton
    );

    // Set the scene for the main window
    Scene scene = new Scene(layout, 300, 250);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

```
    // Methods to handle button clicks for each section
    private void openPatientRegistration() {
        PatientRegistration registration = new PatientRegistration();
        registration.show();
    }

    private void openBilling() {
        Billing billing = new Billing();
        billing.show();
    }

    private void openInvoices() {
        Invoices invoices = new Invoices();
        invoices.show();
    }

    private void openMedications() {
        Medications medications = new Medications();
        medications.show();
    }

    private void openServices() {
        Services services = new Services();
        services.show();
    }

    private void openPayments() {
        Payments payments = new Payments();
        payments.show();
    }

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

3.3 HOME PAGE DESIGN :

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.image.Image;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundImage;
import javafx.scene.layout.BackgroundPosition;
import javafx.scene.layout.BackgroundRepeat;
import javafx.scene.layout.BackgroundSize;
import javafx.scene.layout.StackPane;
```

```
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;

public class Home extends Application {

    @Override
    public void start(Stage primaryStage) {
        // Call method to open the main window after successful login
        openMainWindow(primaryStage);
    }

    // Method to open the main window with buttons for different sections
    private void openMainWindow(Stage primaryStage) {
        // Create buttons for different sections (Patient Registration, Billing, etc.)
        Button patientRegistrationButton = new Button("Patient Registration");
        Button billingButton = new Button("Billing");
        Button invoiceButton = new Button("Invoices");
        Button medicationButton = new Button("Medications");
        Button servicesButton = new Button("Services");
        Button paymentsButton = new Button("Payments");

        // Set button styles (fonts, size, etc.)
        patientRegistrationButton.setFont(new Font("Arial", 14));
        billingButton.setFont(new Font("Arial", 14));
        invoiceButton.setFont(new Font("Arial", 14));
        medicationButton.setFont(new Font("Arial", 14));
        servicesButton.setFont(new Font("Arial", 14));
        paymentsButton.setFont(new Font("Arial", 14));

        // Event handlers for buttons to open the corresponding sections
        patientRegistrationButton.setOnAction(e -> openPatientRegistration());
        billingButton.setOnAction(e -> openBilling());
        invoiceButton.setOnAction(e -> openInvoices());
        medicationButton.setOnAction(e -> openMedications());
        servicesButton.setOnAction(e -> openServices());
        paymentsButton.setOnAction(e -> openPayments());

        // Layout for the main window with vertical arrangement of buttons
        VBox layout = new VBox(20); // Adjust spacing between buttons
        layout.getChildren().addAll(
                patientRegistrationButton,
                billingButton,
                invoiceButton,
                medicationButton,
                servicesButton,
                paymentsButton
        );
        layout.setAlignment(Pos.CENTER); // Center-align buttons
```

```
    // Set background image
    StackPane root = new StackPane();
    BackgroundImage backgroundImage = new BackgroundImage(
            new Image("file:/D:/image.jpg"),  // Replace with your image path
            BackgroundRepeat.NO_REPEAT,
            BackgroundRepeat.NO_REPEAT,
            BackgroundPosition.CENTER,
            BackgroundSize.DEFAULT
    );
    root.setBackground(new Background(backgroundImage));
    // Add layout to the root pane
    root.getChildren().add(layout);

    // Set the scene for the main window
    Scene scene = new Scene(root, 600, 400);  // Increase window size for better visibility
    primaryStage.setTitle("Hospital Management System");
    primaryStage.setScene(scene);
    primaryStage.show();
}
// Methods to handle button clicks for each section
private void openPatientRegistration() {
    // Logic to open the Patient Registration page
    System.out.println("Patient Registration Page Opened");
}

private void openBilling() {
    // Logic to open the Billing page
    System.out.println("Billing Page Opened");
}

private void openInvoices() {
    // Logic to open the Invoices page
    System.out.println("Invoices Page Opened");
}

private void openMedications() {
    // Logic to open the Medications page
    System.out.println("Medications Page Opened");
}

private void openServices() {
    // Logic to open the Services page
    System.out.println("Services Page Opened");
}

private void openPayments() {
    // Logic to open the Payments page
    System.out.println("Payments Page Opened");
}
```

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

3.4 PATIENT REGISTRATION PAGE DESIGN

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class PatientRegistration extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Patient Registration");

        // Create the root layout (GridPane)
        GridPane grid = new GridPane();
        grid.setAlignment(Pos.CENTER);
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));

        // Set the background image
        grid.setStyle("-fx-background-image: url(file:/D:/image.jpg);"
                + "-fx-background-size: cover; -fx-background-position: center;");

        // Create and add form fields for patient registration
        Label nameLabel = new Label("Patient Name:");
        nameLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(nameLabel, 0, 0);

        TextField nameField = new TextField();
        nameField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(nameField, 1, 0);

        Label ageLabel = new Label("Age:");
        ageLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
```

```
        grid.add(ageLabel, 0, 1);

        TextField ageField = new TextField();
        ageField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill: black;");
        grid.add(ageField, 1, 1);

        Label genderLabel = new Label("Gender:");
        genderLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(genderLabel, 0, 2);

        ComboBox<String> genderComboBox = new ComboBox<>();
        genderComboBox.getItems().addAll("Male", "Female", "Other");
        genderComboBox.setValue("Male");  // Default value
        genderComboBox.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-
fill: black;");
        grid.add(genderComboBox, 1, 2);

        Label contactLabel = new Label("Contact Number:");
        contactLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(contactLabel, 0, 3);

        TextField contactField = new TextField();
        contactField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(contactField, 1, 3);

        // Create an address field
        Label addressLabel = new Label("Address:");
        addressLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(addressLabel, 0, 4);

        TextField addressField = new TextField();
        addressField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
black;");
        grid.add(addressField, 1, 4);

        // Create a medical history field
        Label medicalHistoryLabel = new Label("Medical History:");
        medicalHistoryLabel.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-
fill: black;");
        grid.add(medicalHistoryLabel, 0, 5);

        TextField medicalHistoryField = new TextField();
        medicalHistoryField.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-
fill: black;");
        grid.add(medicalHistoryField, 1, 5);
```

```
        // Register button setup
        Button registerButton = new Button("Register");
        registerButton.setStyle("-fx-font-family: 'Roboto'; -fx-font-weight: bold; -fx-text-fill:
white; -fx-background-color: black;");
        grid.add(registerButton, 1, 6);

        // Set action for Register button
        registerButton.setOnAction(e -> {
            String name = nameField.getText();
            int age = Integer.parseInt(ageField.getText());
            String gender = genderComboBox.getValue();
            String contactInfo = contactField.getText();
            String address = addressField.getText();
            String medicalHistory = medicalHistoryField.getText();

            // Insert the patient into the database
            String query = "INSERT INTO patients (name, age, gender, contact_info, address,
medical_history) VALUES (?, ?, ?, ?, ?, ?)";
            try (Connection connection = DatabaseConnection.connect();
                PreparedStatement stmt = connection.prepareStatement(query)) {

                stmt.setString(1, name);
                stmt.setInt(2, age);
                stmt.setString(3, gender);
                stmt.setString(4, contactInfo);
                stmt.setString(5, address);
                stmt.setString(6, medicalHistory);

                int rowsAffected = stmt.executeUpdate();
                System.out.println(rowsAffected + " patient(s) added.");

                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                alert.setTitle("Registration Successful");
                alert.setHeaderText("Patient Registered Successfully");
                alert.setContentText("The patient has been added to the system.");
                alert.showAndWait();

                // Clear the fields after registration
                nameField.clear();
                ageField.clear();
                contactField.clear();
                addressField.clear();
                medicalHistoryField.clear();

            } catch (SQLException ex) {
                System.out.println("Error inserting patient: " + ex.getMessage());
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Error");
                alert.setHeaderText("Patient Registration Failed");
                alert.setContentText("There was an error while registering the patient.");
```

```
            alert.showAndWait();
        }
    });

    // Create the scene and display it
    Scene scene = new Scene(grid, 400, 400);
    primaryStage.setScene(scene);
    primaryStage.show();
  }

  public static void main(String[] args) {
    launch(args);
  }

  public void show() {
    throw new UnsupportedOperationException("Unimplemented method 'show'");
  }
}
```

3.5 MEDICATIONS PAGE DESIGN

```
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;

public class Medications extends Stage {

  public Medications() {
    setTitle("Medications");

    // Create TextFields with CSS styles
    TextField patientIDField = new TextField();
    patientIDField.setPromptText("Enter Patient ID");
    patientIDField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

    TextField medicationField = new TextField();
    medicationField.setPromptText("Enter Medication");
    medicationField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

    TextField dosageField = new TextField();
```

```
        dosageField.setPromptText("Enter Dosage");
        dosageField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold;
-fx-text-fill: black;");

        // Create Save Button with style
        Button saveButton = new Button("Save Medication");
        saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 16px; -fx-font-weight: bold; -
fx-text-fill: white; -fx-background-color: black;");

        // Set the save action
        saveButton.setOnAction(e -> {
            String patientID = patientIDField.getText();
            String medication = medicationField.getText();
            String dosage = dosageField.getText();

            // Input validation
            if (patientID.isEmpty() || medication.isEmpty() || dosage.isEmpty()) {
                showAlert(AlertType.ERROR, "Validation Error", "Please fill out all fields.");
                return;
            }

            // Save medication to the database
            saveMedicationToDatabase(patientID, medication, dosage);
            showAlert(AlertType.INFORMATION, "Success", "Medication saved successfully.");

            // Clear fields after saving
            patientIDField.clear();
            medicationField.clear();
            dosageField.clear();
        });

        // Layout for the medications form
        VBox layout = new VBox(10);
        layout.getChildren().addAll(patientIDField, medicationField, dosageField, saveButton);

        // Set background image with CSS style
        layout.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
                + "-fx-background-size: cover;");

        // Set the scene and window properties
        Scene scene = new Scene(layout, 400, 300);
        setScene(scene);
    }

    // Database connection details
    private Connection connectToDatabase() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/hospital_management";
        String user = "root";  // replace with your MySQL username
        String password = "kumaran";  // replace with your MySQL password
        return DriverManager.getConnection(url, user, password);
```

```
    }

    // Save medication information to the database
    private void saveMedicationToDatabase(String patientID, String medication, String
dosage) {
        String sql = "INSERT INTO medications (patient_id, medication, dosage) VALUES (?,
?, ?)";

        try (Connection conn = connectToDatabase();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {

            pstmt.setString(1, patientID);
            pstmt.setString(2, medication);
            pstmt.setString(3, dosage);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            showAlert(AlertType.ERROR, "Database Error", "Failed to save medication: " +
e.getMessage());
        }
    }

    // Show alert helper method
    private void showAlert(AlertType alertType, String title, String message) {
        Alert alert = new Alert(alertType);
        alert.setTitle(title);
        alert.setHeaderText(null);
        alert.setContentText(message);
        alert.showAndWait();
    }
}
```

3.6 SERVICES PAGE DESIGN

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javafx.scene.control.Alert;

public class Services extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Services");
```

```
        // Create TextFields with styling
        TextField serviceNameField = new TextField();
        serviceNameField.setPromptText("Enter Service Name");
        serviceNameField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

        TextField descriptionField = new TextField();
        descriptionField.setPromptText("Enter Description");
        descriptionField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

        // Cr`eate Save Button with styling
        Button saveButton = new Button("Save Service");
        saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 16px; -fx-font-weight: bold; -
fx-text-fill: white; -fx-background-color: black;");

        // Layout for the Services form
        VBox layout = new VBox(10);
        layout.getChildren().addAll(serviceNameField, descriptionField, saveButton);

        // Set background image and additional styles
        layout.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
                + "-fx-background-size: cover;");

        // Set the scene and window properties
        Scene scene = new Scene(layout, 400, 300);
        primaryStage.setScene(scene);
        primaryStage.show();

        // Set action for Save Service button
        saveButton.setOnAction(e -> {
            String serviceName = serviceNameField.getText();
            String description = descriptionField.getText();

            // Insert the service into the database
            String query = "INSERT INTO services (service_name, description) VALUES (?, ?)";
            try (Connection connection = DatabaseConnection.connect();
                 PreparedStatement stmt = connection.prepareStatement(query)) {

                stmt.setString(1, serviceName);
                stmt.setString(2, description);

                int rowsAffected = stmt.executeUpdate();
                System.out.println(rowsAffected + " service(s) added.");

                // Display success message
                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                alert.setTitle("Service Registration");
                alert.setHeaderText("Service Saved Successfully");
                alert.setContentText("The service has been added to the system.");
```

```
            alert.showAndWait();

            // Clear fields after saving
            serviceNameField.clear();
            descriptionField.clear();

        } catch (SQLException ex) {
            System.out.println("Error inserting service: " + ex.getMessage());
            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setHeaderText("Service Registration Failed");
            alert.setContentText("There was an error while saving the service.");
            alert.showAndWait();
        }
    });
}

public static void main(String[] args) {
    launch(args);
}

// Database connection class
public static class DatabaseConnection {
    public static Connection connect() throws SQLException {
        // Replace with your MySQL connection details
        String url = "jdbc:mysql://localhost:3306/your_database";
        String username = "root";
        String password = "kumaran";

        return java.sql.DriverManager.getConnection(url, username, password);
    }
}

public void show() {
    throw new UnsupportedOperationException("Unimplemented method 'show'");
}
}
```

3.7 BILLING PAGE DESIGN

```
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Alert.AlertType;

public class Billing {

   public Billing() {
      Stage billingStage = new Stage();  // Create a new Stage instance for the Billing window
      billingStage.setTitle("Billing");

      // Create TextFields
      TextField patientIDField = new TextField();
      patientIDField.setPromptText("Enter Patient ID");
      patientIDField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

      TextField serviceField = new TextField();
      serviceField.setPromptText("Enter Service");
      serviceField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold;
-fx-text-fill: black;");

      TextField amountField = new TextField();
      amountField.setPromptText("Enter Amount");
      amountField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold;
-fx-text-fill: black;");

      // Create a Save Button
      Button saveButton = new Button("Save Billing");
      saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 16px; -fx-font-weight: bold; -
fx-text-fill: white; -fx-background-color: black;");

      // Set the save action
      saveButton.setOnAction(e -> {
         String patientID = patientIDField.getText();
         String service = serviceField.getText();
         String amountText = amountField.getText();

         // Input validation
         if (patientID.isEmpty() || service.isEmpty() || amountText.isEmpty()) {
            showAlert(AlertType.ERROR, "Validation Error", "Please fill out all fields.");
            return;
         }

         try {
            double amount = Double.parseDouble(amountText);
            saveBillingToDatabase(patientID, service, amount);
            showAlert(AlertType.INFORMATION, "Success", "Billing information saved
successfully.");

            // Clear fields after saving
            patientIDField.clear();
            serviceField.clear();
```

```
            amountField.clear();
        } catch (NumberFormatException ex) {
            showAlert(AlertType.ERROR, "Invalid Input", "Amount must be a valid
number.");
        }
    });

    // Layout for the billing form
    VBox layout = new VBox(10);
    layout.getChildren().addAll(patientIDField, serviceField, amountField, saveButton);

    // Set background image with CSS
    layout.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
            + "-fx-background-size: cover;");

    // Set the scene and window properties
    Scene scene = new Scene(layout, 400, 300);
    billingStage.setScene(scene);
    billingStage.show(); // Show the Billing window
}

// Database connection details
private Connection connectToDatabase() throws SQLException {
    String url = "jdbc:mysql://localhost:3306/hospital_management";
    String user = "root";  // replace with your MySQL username
    String password = "kumaran";  // replace with your MySQL password
    return DriverManager.getConnection(url, user, password);
}

// Save billing information to database
private void saveBillingToDatabase(String patientID, String service, double amount) {
    String sql = "INSERT INTO billing (patient_id, service, amount) VALUES (?, ?, ?)";

    try (Connection conn = connectToDatabase();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(1, patientID);
        pstmt.setString(2, service);
        pstmt.setDouble(3, amount);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        showAlert(AlertType.ERROR, "Database Error", "Failed to save billing information:
" + e.getMessage());
    }
}

// Show alert helper method
private void showAlert(AlertType alertType, String title, String message) {
    Alert alert = new Alert(alertType);
    alert.setTitle(title);
```

```
        alert.setHeaderText(null);
        alert.setContentText(message);
        alert.showAndWait();
    }

    public void show() {
        // Assuming this method is supposed to show the billing window
    }
}
```

3.8 PAYMENTS PAGE DESIGN

```
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class Payments extends Stage {

    public Payments() {
        setTitle("Payments");

        // Create TextFields with styling
        TextField billingIDField = new TextField();
        billingIDField.setPromptText("Enter Billing ID");
        billingIDField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

        TextField amountField = new TextField();
        amountField.setPromptText("Enter Payment Amount");
        amountField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold;
-fx-text-fill: black;");

        TextField dateField = new TextField();
        dateField.setPromptText("Enter Payment Date");
        dateField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold; -
fx-text-fill: black;");

        // Create Save Button with styling
        Button saveButton = new Button("Save Payment");
        saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 16px; -fx-font-weight: bold; -
fx-text-fill: white; -fx-background-color: black;");

        // Action for Save Button
```

```
        saveButton.setOnAction(e -> savePayment(billingIDField.getText(),
amountField.getText(), dateField.getText()));

        // Layout for the payment form
        VBox layout = new VBox(10);
        layout.getChildren().addAll(billingIDField, amountField, dateField, saveButton);

        // Set background image and styling
        layout.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
                + "-fx-background-size: cover;");

        // Set the scene and window properties
        Scene scene = new Scene(layout, 400, 300);
        setScene(scene);
    }

    // Method to save payment information to the database
    private void savePayment(String billingID, String amount, String date) {
        // SQL query to insert payment details
        String query = "INSERT INTO payments (billing_id, amount, payment_date) VALUES
(?, ?, ?)";

        // Connect to the database and execute the query
        try (Connection connection = DatabaseConnection.connect();
             PreparedStatement statement = connection.prepareStatement(query)) {

            statement.setString(1, billingID);
            statement.setString(2, amount);
            statement.setString(3, date);

            int rowsAffected = statement.executeUpdate();
            if (rowsAffected > 0) {
                showAlert(Alert.AlertType.INFORMATION, "Success", "Payment saved
successfully.");
            }

        } catch (SQLException e) {
            showAlert(Alert.AlertType.ERROR, "Error", "Failed to save payment: " +
e.getMessage());
        }
    }

    // Helper method to display alerts
    private void showAlert(Alert.AlertType alertType, String title, String message) {
        Alert alert = new Alert(alertType);
        alert.setTitle(title);
        alert.setContentText(message);
        alert.showAndWait();
    }
}
```

3.9 INVOICE PAGE DESIGN

```
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;

public class Invoices extends Stage {

   public Invoices() {
      setTitle("Invoices");

      // Create TextFields with CSS styles
      TextField billingIDField = new TextField();
      billingIDField.setPromptText("Enter Billing ID");
      billingIDField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight:
bold; -fx-text-fill: black;");

      TextField dateField = new TextField();
      dateField.setPromptText("Enter Date (YYYY-MM-DD)");
      dateField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold; -
fx-text-fill: black;");

      TextField totalField = new TextField();
      totalField.setPromptText("Enter Total");
      totalField.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-font-weight: bold; -
fx-text-fill: black;");

      // Create Save Button with style
      Button saveButton = new Button("Save Invoice");
      saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 16px; -fx-font-weight: bold; -
fx-text-fill: white; -fx-background-color: black;");

      // Set the save action
      saveButton.setOnAction(e -> {
         String billingID = billingIDField.getText();
         String date = dateField.getText();
         String totalText = totalField.getText();

         // Input validation
         if (billingID.isEmpty() || date.isEmpty() || totalText.isEmpty()) {
```

```
            showAlert(AlertType.ERROR, "Validation Error", "Please fill out all fields.");
            return;
        }

        try {
            double total = Double.parseDouble(totalText);
            saveInvoiceToDatabase(billingID, date, total);
            showAlert(AlertType.INFORMATION, "Success", "Invoice saved successfully.");

            // Clear fields after saving
            billingIDField.clear();
            dateField.clear();
            totalField.clear();
        } catch (NumberFormatException ex) {
            showAlert(AlertType.ERROR, "Invalid Input", "Total must be a valid number.");
        }
    });

    // Layout for the invoice form
    VBox layout = new VBox(10);
    layout.getChildren().addAll(billingIDField, dateField, totalField, saveButton);

    // Set background image with CSS style
    layout.setStyle("-fx-background-image: url('file:/D:/image.jpg');"
            + "-fx-background-size: cover;");

    // Set the scene and window properties
    Scene scene = new Scene(layout, 400, 300);
    setScene(scene);
}

// Database connection details
private Connection connectToDatabase() throws SQLException {
    String url = "jdbc:mysql://localhost:3306/hospital_management";
    String user = "root";  // replace with your MySQL username
    String password = "kumaran";  // replace with your MySQL password
    return DriverManager.getConnection(url, user, password);
}

// Save invoice information to the database
private void saveInvoiceToDatabase(String billingID, String date, double total) {
    String sql = "INSERT INTO invoices (billing_id, date, total) VALUES (?, ?, ?)";

    try (Connection conn = connectToDatabase();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(1, billingID);
        pstmt.setString(2, date);
        pstmt.setDouble(3, total);
        pstmt.executeUpdate();
```

```
        } catch (SQLException e) {
            showAlert(AlertType.ERROR, "Database Error", "Failed to save invoice: " +
e.getMessage());
        }
    }

    // Show alert helper method
    private void showAlert(AlertType alertType, String title, String message) {
        Alert alert = new Alert(alertType);
        alert.setTitle(title);
        alert.setHeaderText(null);
        alert.setContentText(message);
        alert.showAndWait();
    }
}
```

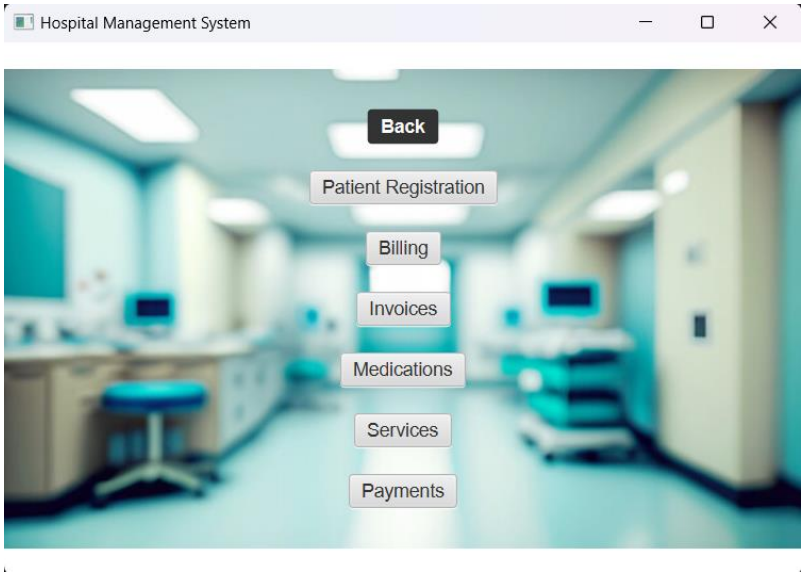
3.10 DATABASE CONNECTIVITY

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/hospital_management";
    private static final String USER = "root";
    private static final String PASSWORD = "kumaran";

    public static Connection connect() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            System.out.println("Database connection error: " + e.getMessage());
            return null;
        }
    }
}
```

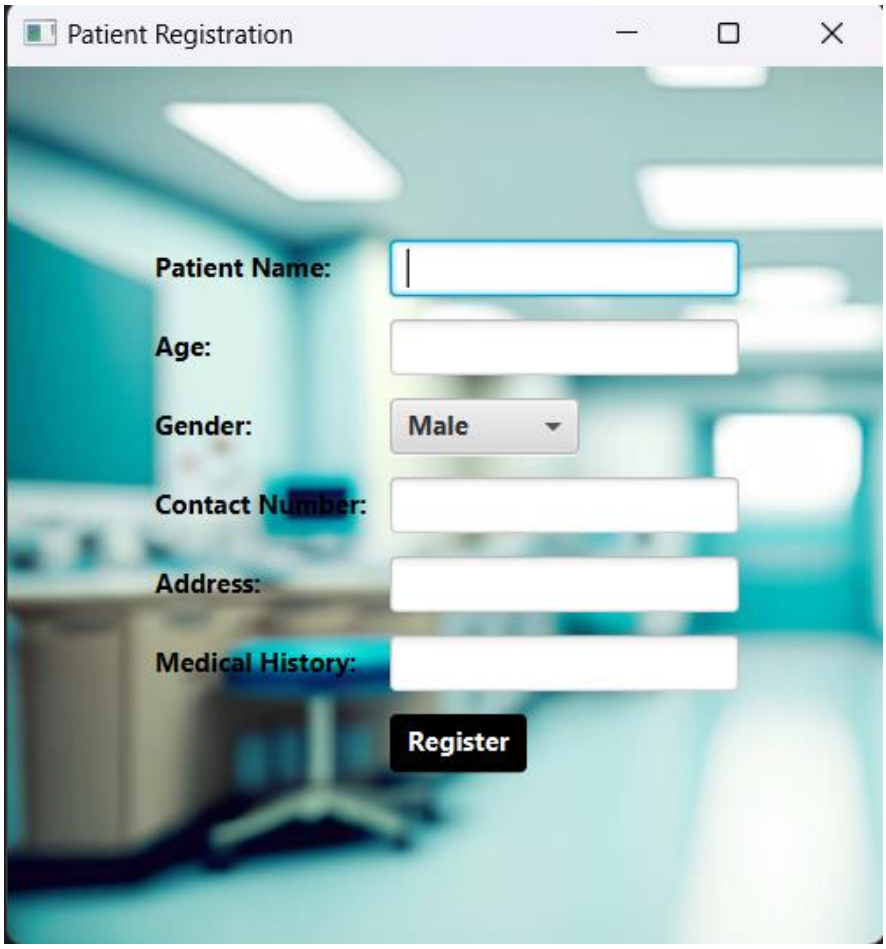
SNAPSHOTS

4.1 LOGIN PAGE:

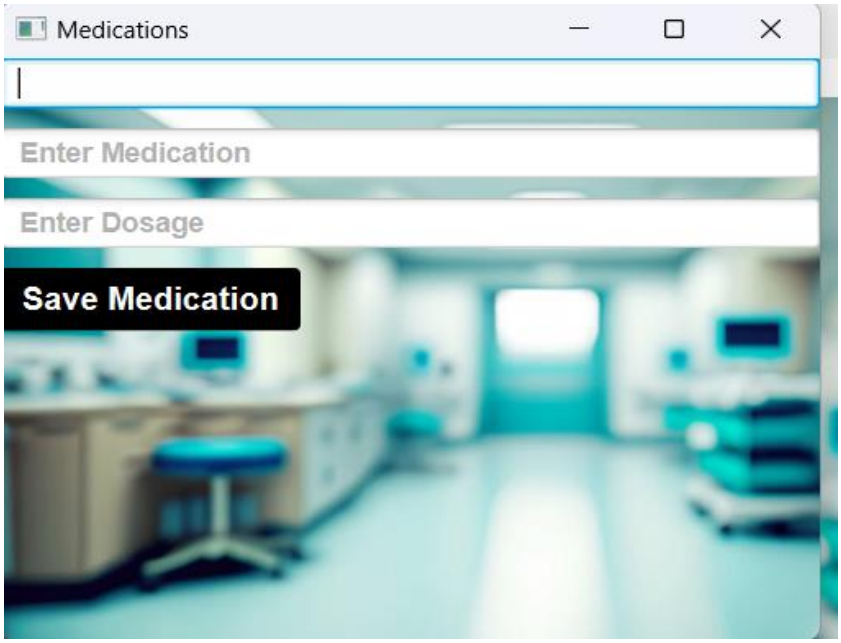
4.2 HOME PAGE:



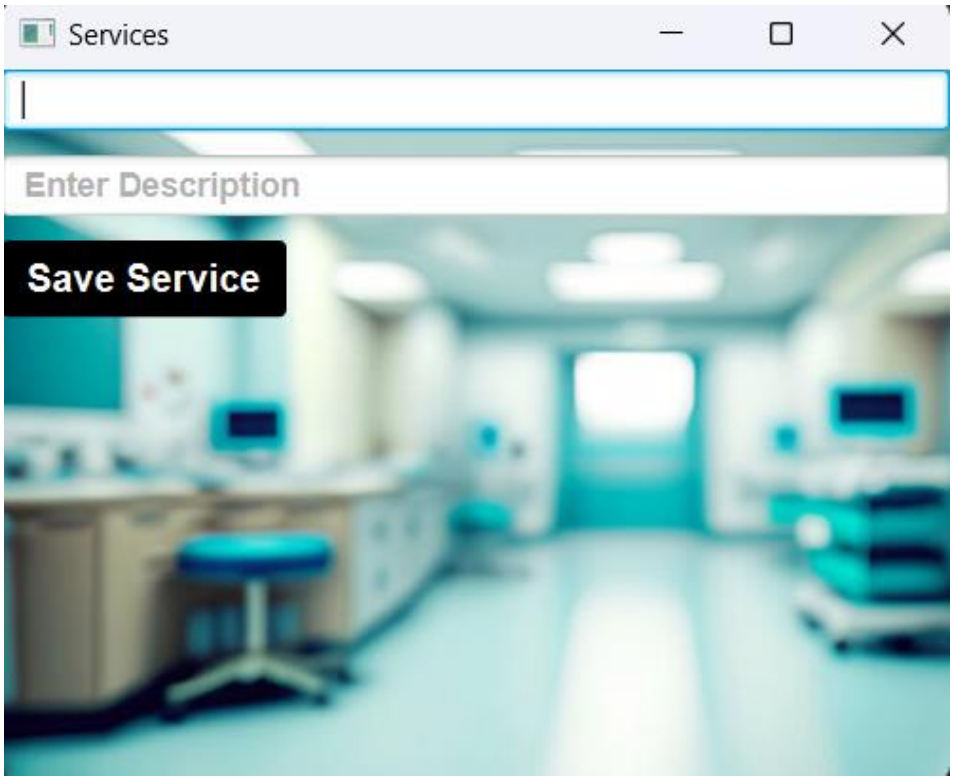
4.3 PATIENT REGISTRATION PAGE:



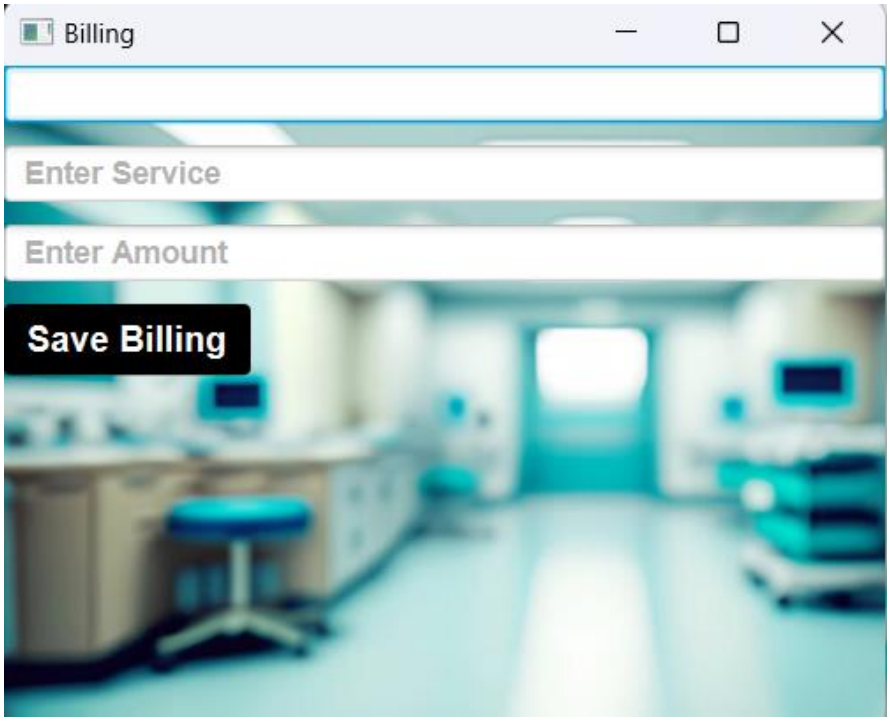
4.4 MEDICATIONS PAGE:



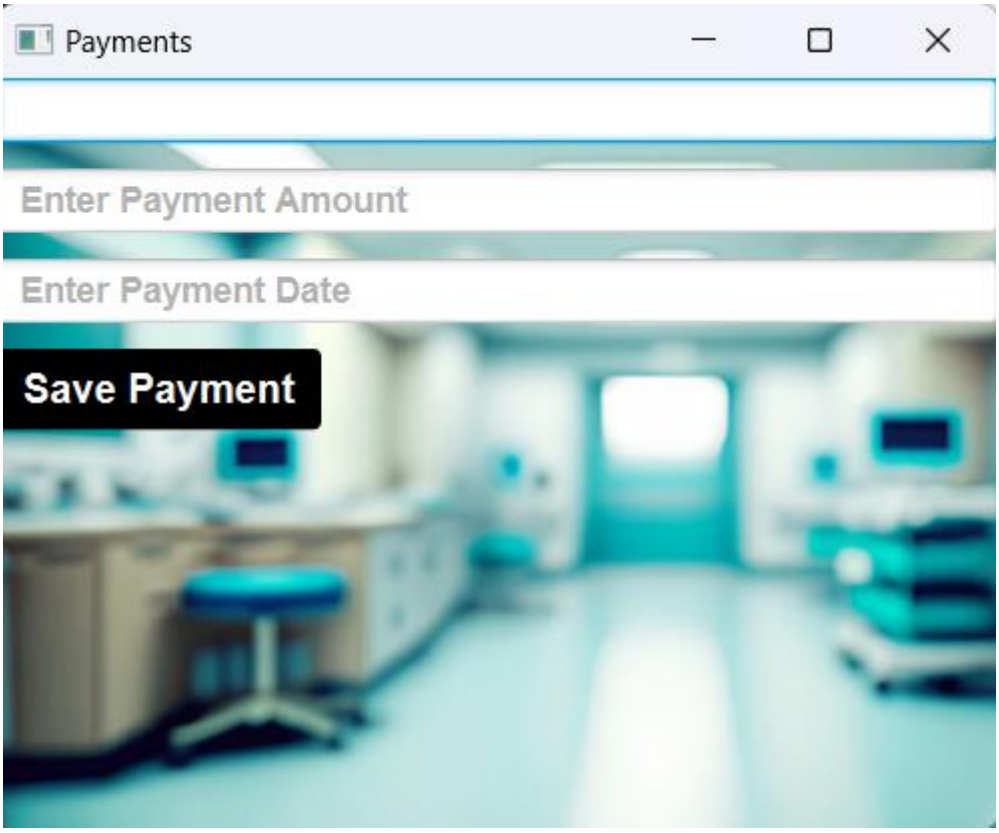
4.5 SERVICES PAGE:



4.6 BILLING PAGE:



4.7 PAYMENTS PAGE:



4.8 INVOICE PAGE: