S.No: 1	Exp. Name: ***Reading data from keyboard and display the data on console***	Date: 2023-08-19

Aim:

Write java programs to demonstrate following

1. Reading data from keyboard and display the data on console.

Source Code:

q209/ReadDataFromKeyboard.java

```
package q209;
import java.util.*;
class ReadDataFromKeyboard{
        public static void main(String []args)
        {

                System.out.print("Enter your name: ");
                Scanner sc = new Scanner(System.in);
                String name = sc.nextLine();
                System.out.print("Enter your age: ");
                int age = sc.nextInt();
                System.out.println("Name: "+name);
                System.out.println("Age: "+age);
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter your name:
Sushi
Enter your age:
20
Name: Sushi
Age: 20

Test Case - 2
User Output
Enter your name:
Rubi
Enter your age:
15
Name: Rubi
Age: 15

S.No: 2	Exp. Name: ***Write the code***	Date: 2023-08-19

Aim:

Write a program to creating and casting of variables in java

Source Code:

ExplicitConversion.java

```
import java.util.*;
class ExplicitConversion
{
        public static void main(String args[])
        {
                System.out.println("Enter float value ");
                Scanner sc = new Scanner(System.in);
                float f = sc.nextFloat();
                int s=(int)f;
                System.out.println("int value = "+s);
                System.out.println("after float widening : "+f);

        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter float value
3.0
int value = 3
after float widening : 3.0

ID: 22B01A12E5

Shri Vishnu Engineering College for Women (Autonomous)

S.No: 3	Exp. Name: ***Use of command line arguments***	Date: 2023-08-19

Aim:

Write java programs to demonstrate following

2. Use of command line arguments.

Source Code:

q210/CommandLineArguments.java

```
package q210;
import java.util.*;
class CommandLineArguments{
        public static void main(String args[]){
                System.out.println("Command Line Arguments:");
                for(int i=0;i<args.length;i++){
                System.out.print("Argument ");
                System.out.print(i+1);
                System.out.println(": "+args[i]);
                }

        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Command Line Arguments:
Argument 1: 5 6 7

Test Case - 2
User Output
Command Line Arguments:
Argument 1: 10
Argument 2: 25
Argument 3: 16

Aim:

Write a Java program that use arithmetic operators, unary operators and conditional operator.

Source Code:

q211/OperatorExample.java

```
package q211;
import java.util.*;
class OperatorExample
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter two numbers for arithmetic operations:");
                System.out.print("Number 1: ");
                int n = sc.nextInt();
                System.out.print("Number 2: ");
                int m = sc.nextInt();
                System.out.print("Sum: ");
                System.out.println(n+m);
                System.out.print("Difference: ");
                System.out.println(n-m);
                System.out.print("Product: ");
                System.out.println(n*m);
                System.out.print("Quotient: ");
                System.out.println(n/m);
                System.out.print("Remainder: ");
                System.out.println(n%m);
                System.out.print("Enter a number to apply unary minus: ");
                int x = sc.nextInt();
                System.out.print("Unary Minus: ");
                System.out.println(-x);
                System.out.print("Enter a number to check if it's even or odd: ");
                int y = sc.nextInt();
                if(y%2==0)
                System.out.print("Number is Even");
                else
                System.out.print("Number is Odd");
                System.out.println();
        }

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter two numbers for arithmetic operations:

Number 1:
5
Number 2:
10
Sum: 15
Difference: -5
Product: 50
Quotient: 0
Remainder: 5
Enter a number to apply unary minus:
6
Unary Minus: -6
Enter a number to check if it's even or odd:
23
Number is Odd

Test Case - 2
User Output
Enter two numbers for arithmetic operations:
Number 1:
-6
Number 2:
8
Sum: 2
Difference: -14
Product: -48
Quotient: 0
Remainder: -6
Enter a number to apply unary minus:
-5
Unary Minus: 5
Enter a number to check if it's even or odd:
22
Number is Even

S.No: 5	Exp. Name: ***Selection Statement - Switch***	Date: 2023-08-19

Aim:

Write Java programs that illustrates Selection statements

Source Code:

q212/SwitchExample.java

```
package q212;
import java.util.*;
class SwitchExample{
        public static void main(String args[]){
                System.out.print("Enter a number between 1 and 7: ");
                Scanner sc = new Scanner(System.in);
                int n = sc.nextInt();
                switch(n)
                {
                        case 1:
                        System.out.println("Sunday");
                        break;
                        case 2:
                        System.out.println("Monday");
                        break;
                        case 3:
                        System.out.println("Tuesday");
                        break;
                        case 4:
                        System.out.println("Wednesday");
                        break;
                        case 5:
                        System.out.println("Thursday");
                        break;
                        case 6:
                        System.out.println("Friday");
                        break;
                        case 7:
                        System.out.println("Saturday");
                        break;

                }


}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a number between 1 and 7:
5
Thursday

Test Case - 2
User Output
Enter a number between 1 and 7:
1
Sunday

S.No: 6	Exp. Name: ***Find the sum of digits of a number***	Date: 2023-08-19

Aim:

Write a java program that finds the sum of digits of a number using loops

Source Code:

q213/loopExample.java

```
package q213;
import java.util.*;
class loopExample{
        public static void main(String args[]){
        System.out.print("Enter a number: ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int sum =0;
        int r;
        int x=n;
        while(n>0)
        {
                r = n % 10;
                sum = sum + r;
                n = n / 10;
        }
        System.out.print("Sum of digits of "+ x +" "+"is: "+sum);
        System.out.println();

        }


}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a number:
1254
Sum of digits of 1254 is: 12

Test Case - 2
User Output
Enter a number:
362145
Sum of digits of 362145 is: 21

S.No: 7	Exp. Name: ***Labelled break and continue statements***	Date: 2023-08-31

Aim:

Write a Java program that uses labeled break and continue statements to demonstrate nested loops. The program should take user input for the number of rows and columns, and then it should print the values of the variables 'i' and 'j' inside the nested loops. The program should use labeled break to exit the nested loop when 'i' is equal to 2 and 'j' is equal to 2, and use labeled continue to skip the iteration when 'i' is equal to 2 and 'j' is equal to 2.

Source Code:

q214/BreakAndContinue.java

```
package q214;
import java.util.*;
class BreakAndContinue{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter the number of rows: ");
                int r=sc.nextInt();
                System.out.print("Enter the number of columns: ");
                int c=sc.nextInt();
                first:
                for(int i=1;i<=r;i++)
                {
                        second:
                        for(int j=1;j<=c;j++)
                        {
                        if(i==2&&j==2)
                        break first;
                        System.out.println("i = "+i+", j = "+j);
                        }
                }
                first: for(int i=1;i<=r;i++){
                        second: for(int j=1;j<=c;j++){
                                if(i==2 && j==2)
                                continue first;
                                System.out.println("i = "+i+", j = "+j);
                        }
                }


        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of rows:
2
Enter the number of columns:

2
i = 1, j = 1
i = 1, j = 2
i = 2, j = 1
i = 1, j = 1
i = 1, j = 2
i = 2, j = 1

Test Case - 2
User Output
Enter the number of rows:
3
Enter the number of columns:
4
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
i = 1, j = 4
i = 2, j = 1
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
i = 1, j = 4
i = 2, j = 1
i = 3, j = 1
i = 3, j = 2
i = 3, j = 3
i = 3, j = 4

S.No: 8	Exp. Name: ***1-D array***	Date: 2023-08-19

Aim:

Write a program to demonstrate 1-D array.

Source Code:

q215/OneDArray.java

```
package q215;
import java.util.*;
class OneDArray{
        public static void main(String args[]){
                System.out.print("Enter the size of the array: ");
                Scanner sc = new Scanner(System.in);
                int n = sc.nextInt();
                int []a=new int[20];
                int sum =0;
                System.out.println("Enter "+n+" elements:");
                for(int i =0;i<n;i++)
                {
                  a[i]=sc.nextInt();

                }
                System.out.println("The elements of the array are:");
                for(int i=0;i<n;i++)
                {
                        System.out.print(a[i]+" ");
                }
                System.out.println();
                for(int i=0;i<n;i++)
                {
                        sum = sum + a[i];
                }
                System.out.println("The sum of all elements in the array is: "+sum);
                int max =a[0];
                int min=a[0];
                for(int i=0;i<n;i++)
                {
                        if(a[i]>max)
                        max=a[i];
                        if(a[i]<min)
                        min=a[i];

                }
                System.out.println("The maximum element in the array is: "+max);
                System.out.println("The minimum element in the array is: "+min);



        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the size of the array:
5
Enter 5 elements:
3 6 8 9 4
The elements of the array are:
3 6 8 9 4
The sum of all elements in the array is: 30
The maximum element in the array is: 9
The minimum element in the array is: 3

Test Case - 2
User Output
Enter the size of the array:
6
Enter 6 elements:
1 2 3 4 5 6
The elements of the array are:
1 2 3 4 5 6
The sum of all elements in the array is: 21
The maximum element in the array is: 6
The minimum element in the array is: 1

S.No: 9	Exp. Name: ***Iterating through a multidimensional array.***	Date: 2023-08-19

Aim:

The class `MultiDimArrayPrinter` prints a multidimensional array of integers.

An integer two-dimensional array **int2DArr** is declared and initialized in the given code. Your task is to :

Take an integer **n** from the user.

Use a nested for loop to iterate over the first **n** rows of the **int2DArr** array

Note:

- The code provided in the editor reads the input integer using the nextInt() method and stores it in **n** variable.
- Constraints: n < 4
- Please don't change the package name.

Source Code:

q10947/MultiDimArrayPrinter.java

```
package q10947;
import java.util.*;
public class MultiDimArrayPrinter {
        public static void main(String[] args) {
                int[][] int2DArr = {
                                                        {1},
                                                        {2, 3},
                                                        {4, 5, 6},
                                                        {7, 8, 9, 10}
                                                };
                Scanner sc = new Scanner(System.in);
                int n = sc.nextInt();

        //Write your code here....
        int num=1;
        for(int i =1;i<=n;i++)
        {
                for(int j=1;j<=i;j++)
                {
                        System.out.print(num+" ");
                        num++;
                }
                System.out.println();
        }
}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
2

1
2 3

Test Case - 2
User Output
3
1
2 3
4 5 6

Test Case - 3
User Output
1
1

S.No: 10	Exp. Name: ***Program to find Addition of Two matrices***	Date: 2023-08-31

Aim:

Write a class `AdditionOfMatrix` with a **public** method `add` which returns the sum of its arguments. If the two arguments are not matrices of the same size, then the method should return null.

Consider the following examples for your understanding:

```
Matrix 1:
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 3 4
Matrix 2:
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 4 5
Enter row 2: 6 7
Sum of the two given matrices is:
5 7
9 11
```

```
Matrix 1:
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 3 4
Matrix 2:
Enter number of rows: 2
Enter number of columns: 3
Enter 3 numbers separated by space
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Addition of different sized matrices is not possible
```

Note: Please don't change the package name.

Source Code:

`q11053/AdditionOfMatrix.java`

ID: 22B01A12E5

Shri Vishnu Engineering College for Women (Autonomous)

```
package q11053;
public class AdditionOfMatrix {

	/**
	 * Computes the addition of the two given matrices if the sizes of the matrices are
the same.
	 * Otherwise, returns null.
	 *
	 * @return the resultant matrix, null if addition is not possible
	 */
	public int[][] add(int[][] matrix1, int[][] matrix2) {
		// Write the code
		int rows = matrix1.length;
		int cols =matrix1[0].length;
		int r1 =matrix2.length;
		int c1 =matrix2[0].length;
		if(rows==r1 && cols==c1){
		int[][] c = new int [rows][cols];
		for(int i=0;i<rows;i++){
			for(int j=0;j<cols;j++){
				c[i][j] = matrix1[i][j]+matrix2[i][j]
;			}
		}
		return c;
		}
		else
		return null;

	}
}
```

q11053/AdditionOfMatrixMain.java

```
package q11053;
import java.util.Scanner;

public class AdditionOfMatrixMain {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		AdditionOfMatrix adder = new AdditionOfMatrix();

		System.out.println("Matrix 1:");
		int[][] m1 = readMatrix(s);

		System.out.println("Matrix 2:");
		int[][] m2 = readMatrix(s);

		int[][] sum = adder.add(m1, m2);

		if (sum == null) {
			System.out.println("Addition of different sized matrices is not
possible");
		} else {
			System.out.println("Sum of the two given matrices is:");
			for (int i = 0; i < sum.length; i++) {
				int c = sum[i].length;
				for (int j = 0; j < c; j++) {
					String spacer = j == c - 1 ? "\n" : " ";
					System.out.print(sum[i][j] + spacer);
				}
			}
		}
	}

	public static int[][] readMatrix(Scanner s) {
		System.out.print("Enter number of rows: ");
		int r = s.nextInt();
		System.out.print("Enter number of columns: ");
		int c = s.nextInt();
		int[][] m = new int[r][c];
		System.out.println("Enter " + c + " numbers separated by space");
		for (int i = 0; i < r; i++) {
			System.out.print("Enter row " + (i + 1) + ": ");
			for (int j = 0; j < c; j++) {
				m[i][j] = s.nextInt();
			}
		}
		return m;
	}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Matrix 1:

```
Enter number of rows:
2
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
1 2
Enter row 2:
3 4
Matrix 2:
Enter number of rows:
2
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
4 5
Enter row 2:
6 7
Sum of the two given matrices is:
5 7
9 11
```

S.No: 11	Exp. Name: ***Program to find Multiplication of Two matrices***	Date: 2023-09-21

Aim:

Write a class `MultiplicationOfMatrix` with a **public** method `multiplication` which returns the multiplication result of its arguments. if the first argument column size is not equal to the row size of the second argument, then the method should return null.

Consider the following example for your understanding

```
Matrix 1:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 4 5
Enter row 3: 7 8
Matrix 2:
Enter number of rows: 2
Enter number of columns: 3
Enter 3 numbers separated by space
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Multiplication of the two given matrices is:
9 12 15
24 33 42
39 54 69
```

```
Matrix 1:
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 3 4
Matrix 2:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 4 5
Enter row 3: 2 3
Multiplication of matrices is not possible
```

Note: Please don't change the package name.

Source Code:

`q11106/MultiplicationOfMatrix.java`

```
package q11106;
public class MultiplicationOfMatrix{
	public int[][] multiplication(int[][] matrix1, int[][] matrix2) {
		/*Return the result if the matrix1 coloumn size is equal to matrix2 row size
and print the result.
		* @Return null.
		*/
		// Write your logic here for matrix multiplication
		int r1=matrix1.length;
		int r2=matrix2.length;
		int c1=matrix1[0].length;
		int c2=matrix2[0].length;
		if(c1==r2){
			int [][] c= new int[r1][c2];
			for(int i=0;i<r1;i++){
				for(int j=0;j<c2;j++){
					c[i][j]=0;
				for(int k=0;k<c1;k++){
					c[i][j]+=matrix1[i][k]*matrix2[k][j];
				}
				}
			}
			return c;
		}
		else
		return null;
	}
}
```

q11106/MultiplicationOfMatrixMain.java

```
package q11106;
import java.util.Scanner;
public class MultiplicationOfMatrixMain {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		MultiplicationOfMatrix multiplier = new MultiplicationOfMatrix();

		System.out.println("Matrix 1:");
		int[][] m1 = readMatrix(s);

		System.out.println("Matrix 2:");
		int[][] m2 = readMatrix(s);

		int[][] multi = multiplier.multiplication(m1, m2);

		if (multi == null) {
			System.out.println("Multiplication of matrices is not possible");
		} else {
			System.out.println("Multiplication of the two given matrices is:");
			for (int i = 0; i < multi.length; i++) {
				int c = multi[i].length;
				for (int j = 0; j < c; j++) {
					String spacer = j == c - 1 ? "\n" : " ";
					System.out.print(multi[i][j] + spacer);
				}
			}
		}
	}

	public static int[][] readMatrix(Scanner s) {
		System.out.print("Enter number of rows: ");
		int r = s.nextInt();
		System.out.print("Enter number of columns: ");
		int c = s.nextInt();
		int[][] m = new int[r][c];
		System.out.println("Enter " + c + " numbers separated by space");
		for (int i = 0; i < r; i++) {
			System.out.print("Enter row " + (i + 1) + ": ");
			for (int j = 0; j < c; j++) {
				m[i][j] = s.nextInt();
			}
		}
		return m;
	}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Matrix 1:
Enter number of rows:
2

Enter number of columns:
3
Enter 3 numbers separated by space
Enter row 1:
1 2 3
Enter row 2:
4 5 6
Matrix 2:
Enter number of rows:
3
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
1 2
Enter row 2:
3 4
Enter row 3:
5 6
Multiplication of the two given matrices is:
22 28
49 64

Test Case - 2
User Output
Matrix 1:
Enter number of rows:
2
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
1 2
Enter row 2:
3 4
Matrix 2:
Enter number of rows:
2
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
5 6
Enter row 2:
7 8
Multiplication of the two given matrices is:
19 22
43 50

S.No: 12	Exp. Name: ***Write a Java program to display Transpose of the given Matrix***	Date: 2023-08-31

Aim:

Write a class `TransposeMatrix` with a **public** method `transposeMatrix` that takes one parameter `matrix1` of type `int[][]` which returns the transpose of the given matrix.

Consider the following example for your undertanding:

```
Matrix:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 3 4
Enter row 3: 5 6
Transpose of given matrices is:
1 3 5
2 4 6
```

Note: Please don't change the package name.

Source Code:

q11105/TransposeMatrix.java

```
package q11105;
public class TransposeMatrix {
/**
         * Computes the transpose of the given matrix.
         *
         * @return the resultant matrix
         */

public int[][] transposeMatrix(int[][] matrix1) {

                // Write the code
                int r1 = matrix1.length;
                int c1 = matrix1[0].length;
                int [][] c = new int [c1][r1];
                for(int i=0;i<r1;i++){
                        for(int j=0;j<c1;j++){
                                c[j][i] = matrix1[i][j];
                        }
                }
                return c;

        }
}
```

q11105/TransposeMain.java

```
package q11105;
import java.util.Scanner;
public class TransposeMain {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		TransposeMatrix tr = new TransposeMatrix();

		System.out.println("Matrix:");
		int[][] m1 = readMatrix(s);

		int[][] transpose = tr.transposeMatrix(m1);

		if (transpose == null) {
			System.out.println("Transpose of matrix is not possible");
		} else {
			System.out.println("Transpose of given matrices is:");
			for (int i = 0; i < transpose.length; i++) {
				int c = transpose[i].length;
				for (int j = 0; j < c; j++) {
					String spacer = j == c - 1 ? "\n" : " ";
					System.out.print(transpose[i][j] + spacer);
				}
			}
		}
	}

	public static int[][] readMatrix(Scanner s) {
		System.out.print("Enter number of rows: ");
		int r = s.nextInt();
		System.out.print("Enter number of columns: ");
		int c = s.nextInt();
		int[][] m = new int[r][c];
		System.out.println("Enter " + c + " numbers separated by space");
		for (int i = 0; i < r; i++) {
			System.out.print("Enter row " + (i + 1) + ": ");
			for (int j = 0; j < c; j++) {
				m[i][j] = s.nextInt();
			}
		}
		return m;
	}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Matrix:
Enter number of rows:
3
Enter number of columns:
2

Enter 2 numbers separated by space
Enter row 1:
1 2
Enter row 2:
3 4
Enter row 3:
5 6
Transpose of given matrices is:
1 3 5
2 4 6

Test Case - 2
User Output
Matrix:
Enter number of rows:
2
Enter number of columns:
2
Enter 2 numbers separated by space
Enter row 1:
1 2
Enter row 2:
62 32
Transpose of given matrices is:
1 62
2 32

S.No: 13	Exp. Name: ***Implement a class with data members and methods***	Date: 2023-08-22

Aim:

Write a Java program to implement a class with data members and methods

Source Code:

q216/RectangleClass.java

```
package q216;
import java.util.*;
class RectangleClass{
        public static void main(String args[]){
                System.out.print("Enter length of the rectangle: ");
                Scanner sc = new Scanner(System.in);
                Float l = sc.nextFloat();
                System.out.print("Enter width of the rectangle: ");
                Float w = sc.nextFloat();
                System.out.println("Rectangle Details:");
                System.out.println("Length: "+l);
                System.out.println("Width: "+w);
                System.out.println("Area: "+l*w);
                System.out.println("Perimeter: "+ 2*(l+w));
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter length of the rectangle:
5
Enter width of the rectangle:
4
Rectangle Details:
Length: 5.0
Width: 4.0
Area: 20.0
Perimeter: 18.0

Test Case - 2
User Output
Enter length of the rectangle:
58
Enter width of the rectangle:
1
Rectangle Details:

Length: 58.0
Width: 1.0
Area: 58.0
Perimeter: 118.0

S.No: 14	Exp. Name: ***Program to use this keyword***	Date: 2023-08-22

Aim:

Write a Java program to give an example of "**this**" operator. And also use the "**this**" keyword as return statement. Create a class `ThisDemo` with a main method. In the main method, create an object to the class `Rectangle`. Create another class `Rectangle`

- contains the data members **length** and **breadth** of int data type.
- write a method **show()** to initialize the data members.
- write a method **calculate()** which should return the area of rectangle.

Sample Input and Sample Output:

```
Enter length and breadth of rectangle: 5 6
The area of a Rectangle is: 30
```

Source Code:

q226/ThisDemo.java

```
package q226;
import java.util.*;

class Rectangle{
        int l;
        int b;
        public void show(int l,int b){
                this.l=l;
                this.b=b;
        }
        public int calculate(){
                return this.l*this.b;
        }
}
class ThisDemo{
        public static void main(String args[]){
                System.out.print("Enter length and breadth of rectangle: ");
                Scanner sc = new Scanner(System.in);
                int l = sc.nextInt();
                int b = sc.nextInt();
                Rectangle rec = new Rectangle();
                rec.show(l,b);
                int a=rec.calculate();
                System.out.println("The area of a Rectangle is: "+a);
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter length and breadth of rectangle:
5 6

The area of a Rectangle is: 30

Test Case - 2
User Output
Enter length and breadth of rectangle:
3 10
The area of a Rectangle is: 30

S.No: 15	Exp. Name: ***String and String Buffer classes***	Date: 2023-10-11

Aim:

Write a program that illustrates String and String Buffer classes

For string, perform the operations concatenation and find the length of the new string.
For StringBuffer, insert an user input string at a specified index and perform the reverse operation.

Source Code:

q217/StringDemo.java

```
package q217;
import java.util.*;
class StringDemo{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter first string: ");
                String s1 = sc.nextLine();
                System.out.print("Enter second string: ");
                String s2 = sc.nextLine();
                String s=s1+" "+s2;
                int l=s.length();
                System.out.println("Concatenated String: "+s);
                System.out.println("Length of Concatenated String: "+l);
                StringBuffer sb=new StringBuffer();
                System.out.print("Enter a string for StringBuffer: ");
                String x=sc.nextLine();
                System.out.print("Enter a string to append: ");
                String y=sc.nextLine();
                sb.append(x+y);
                System.out.println("Appended StringBuffer: "+sb);
                System.out.print("Enter the index to insert: ");
                int i=sc.nextInt();
                sc.nextLine();
                System.out.print("Enter a string to insert: ");
                String z=sc.nextLine();
                sb.insert(i,z);
                System.out.println("Inserted StringBuffer: "+sb);
                sb.reverse();
                System.out.println("Reversed StringBuffer: "+sb);

        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter first string:
Coding
Enter second string:

in Java
Concatenated String: Coding in Java
Length of Concatenated String: 14
Enter a string for StringBuffer:
Programming
Enter a string to append:
Language
Appended StringBuffer: ProgrammingLanguage
Enter the index to insert:
10
Enter a string to insert:
@
Inserted StringBuffer: Programmin@gLanguage
Reversed StringBuffer: egaugnaLg@nimmargorP

Test Case - 2
User Output
Enter first string:
Progamming
Enter second string:
Error
Concatenated String: Progamming Error
Length of Concatenated String: 16
Enter a string for StringBuffer:
ListIndexError
Enter a string to append:
OutOfBounds
Appended StringBuffer: ListIndexErrorOutOfBounds
Enter the index to insert:
1
Enter a string to insert:
..
Inserted StringBuffer: L..istIndexErrorOutOfBounds
Reversed StringBuffer: sdnuoBfOtuOrorrExednItsi..L

S.No: 16	Exp. Name: ***Constructors with parameters***	Date: 2023-09-21

Aim:

Demonstrate the Constructors with parameters

You are tasked with writing a Java program that demonstrates the concept of constructors with parameters. In this program, you will create a class **Student**, which represents a student with the following attributes:

- name: A string that stores the name of the student.
- age: An integer that stores the age of the student.
- course: A string that stores the name of the course the student is enrolled in.

The Student class will have a parameterized constructor that takes three arguments: name, age, and course. The constructor will initialize the attributes of the **Student** object with the provided values.

Additionally, the Student class will have a method called displayDetails(). This method will display the details of the student, including their name, age, and course.

You need to write the main method in the ConstructorParameters class to interact with the user and demonstrate the use of the parameterized constructor. The main method should perform the following steps:

- Create a Scanner object to read input from the user.
- Prompt the user to enter details for two students, namely "Student 1" and "Student 2".
- For each student, the program should prompt the user to enter the following details:
- Student's name (a string input).Student's age (an integer input).Student's course (a string input).
- Use the provided parameterized constructor to create two Student objects, one for each student, using the input provided by the user

Display the details of both students using the displayDetails() method.

Source Code:

```
q218/ConstructorParameters.java
```

```
package q218;
import java.util.*;
class ConstructorParameters{
        public static void main(String args[]){
                String name1,course1,name2,course2;
                int age1,age2;

                Scanner sc = new Scanner(System.in);
                System.out.print("Enter name of Student 1: ");
                 name1=sc.next();
                System.out.print("Enter age of Student 1: ");
                 age1 = sc.nextInt();
                System.out.print("Enter course of Student 1: ");
            course1 = sc.next();
                System.out.print("Enter name of Student 2: ");
                 name2=sc.next();
                System.out.print("Enter age of Student 2: ");
                 age2=sc.nextInt();
                System.out.print("Enter course of Student 2: ");
                course2=sc.next();
                Student s1 = new Student(name1,age1,course1);
                Student s2 = new Student(name2,age2,course2);
                s1.displayDetails(1);
                s2.displayDetails(2);
        }
}
class Student{
        int i;
        String name,course;
        Student(String name,int i,String course){
                this.name=name;
                this.course=course;
                this.i=i;
        }
        public void displayDetails(int j){
                System.out.println("Details of Student "+j+":");
                System.out.println("Name: "+name+"\n"+"Age: "+i+"\n"+"Course: "+course);
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter name of Student 1:
Joseph
Enter age of Student 1:
21
Enter course of Student 1:
CSE
Enter name of Student 2:
Paul

Enter age of Student 2:
22
Enter course of Student 2:
ECE
Details of Student 1:
Name: Joseph
Age: 21
Course: CSE
Details of Student 2:
Name: Paul
Age: 22
Course: ECE

Test Case - 2
User Output
Enter name of Student 1:
Rosy
Enter age of Student 1:
20
Enter course of Student 1:
MECH
Enter name of Student 2:
Lusy
Enter age of Student 2:
21
Enter course of Student 2:
AI
Details of Student 1:
Name: Rosy
Age: 20
Course: MECH
Details of Student 2:
Name: Lusy
Age: 21
Course: AI

S.No: 17	Exp. Name: ***Constructors without parameters***	Date: 2023-09-21

Aim:

Write a program to demonstrate the Constructors without parameters

1. Write a class named **Student**, which represents a student and has three attributes:
   - name: String (name of the student)
   - age: int (age of the student)
   - course: String (course of the student)

2. The class **Student** has a constructor without parameters:
   - This constructor initializes the attributes name, age, and course with default values "Unknown", 0, and "Unknown" respectively.

3. The class **Student** also has two methods:
   - "setDetails": A method that allows setting the student details (name, age, and course).
   - "displayDetails": A method that displays the student details (name, age, and course).

4. In the main method of the **ConstructorWithoutParameter** class:
   - Create a Scanner object .
   - Create a `Student` object using the constructor without parameters, initializing default values for name, age, and course.
   - Take user input for the student's details and set them using the "setDetails" method.
   - Display the student details using the "displayDetails" method.

Source Code:

q219/ConstructorWithoutParameter.java

```
package q219;
import java.util.*;
class Student{
        String name;
        int age;
        String course;
        public void displayDetails(String name,int age,String course){
                System.out.println("Student Details:");
                System.out.println("Name: "+name);
                System.out.println("Age: "+age);
                System.out.println("Course: "+course);
        }
}
class ConstructorWithoutParameter{
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                System.out.print("Enter name of student: ");
                String name=sc.nextLine();
                System.out.print("Enter age of student: ");
                int age=sc.nextInt();
                System.out.print("Enter course of student: ");
                String  course=sc.next();
                Student t=new Student();
                t.displayDetails(name,age,course);

        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter name of student:
Philips
Enter age of student:
21
Enter course of student:
CSE-AI
Student Details:
Name: Philips
Age: 21
Course: CSE-AI

Test Case - 2
User Output
Enter name of student:
Muskan
Enter age of student:
20
Enter course of student:
CSE-ML
Student Details:
Name: Muskan
Age: 20
Course: CSE-ML