Java Programming Exercise

print, println, Math

1. W.A.P. to print “Welcome 2 Christ College-RAJKOT”.
2. W.A.P. to print “Hello” in four corners of screen and in center of the screen.
3. W.A.P. to print reverse number of given number. Eg: input: 549 output: 945
4. Create class named **MathDemo**. Calculate & print area of circle. Use Math class to get value of PI. (Refer Math class in JavaDocs.)
5. W.A.P. to calculate simple interest and compound interest. (Formula for compound interest A=P(1+R/100)^N)
6. Write a program that prints a conversion table from miles to kilometers. The program should produce the following output to the screen

   miles kilometers
   10.00 16.09
   20.00 32.19

7. Write a program that can convert degrees Fahrenheit to degrees Celsius.
   C= (F-32) * 5 / 9

8. Write a C Program to find area of a rectangle (Hint: A = l*b)
9. Write a C Program to find area of a triangle (Hint: A = ½ *h*b)
10. Write a C Program to find the volume of a cube ($V=l^3$), volume of a cuboid (Hint: V = l*b*h), volume of a cylinder ($V=\Pi r^2h$), volume of a sphere ($V=4/3\Pi r^3$), volume of a cone ($V=1/3\Pi r^2h$)

Class exercises

1. Write a program in Java to check if a number is even or odd in Java? (input 2 output true, input 3 : output false)

2. Write a program in Java to find out if a number is prime in Java? (input 7: output true , input 9 : output false)

3. Write Java program to check if a number is palindrome in Java? (121 is palindrome, 321 is not)

4. How to find if a number is power of 2 in Java? (1,2, 4 power of 2, 3 is not)

5. Write program to sort an integer array without using API methods?

6. Write Java program to check if a number is Armstrong number or not? (input 153 output true, 123 output false)

7. Write a program in Java to reverse any String without using StringBuffer?

11. Write a program in Java to print Fibonacci series up to given number? Write both iterative and recursive version.

12. Write a Java program to calculate factorial of an integer number ? Both iterative and recursive solution.

13. Write a Java program to print primes numbers between two limits using command line arguments

14. Write a Java program Implementing Constructor overloading to find volume of a cube

15. Write a Java program to find area of different shapes using method overriding

16. Write a Java program to calculate salary of different department using abstract class

17. Write a Java program to Perform the String operations

18. Write a Java program to implement thread and thread priorities

19. To perform all the Array functions by using Command Line Arguments.

20. Write a Java program to Print following structure in Java?

```
*
***
*****
***
*
```

Reference variable, object creation, String class, StringBuffer Class.

(String class property: length and methods: trim, toUpperCase, toLowerCase, substring(si), substring(si,ei), startsWith, replaceFirst, replaceAll, replace, length, lastIndexOf,isEmpty,indexOf,equals,equalsIgnoreCase,endsWith, compareTo,compareToIgnoreCase,charAt**)**

(StringBuffer class methods: append, capacity, delete, deleteCharAt, insert, length, replace, reverse, substring, trimToSize**)**

1. Create a class in java named **StringDemo1**. Create 1 String object with initial value "CHRIST". Print each character of this object in new line.
2. Create a class in java named **StringDemo2**. Create 1 String object with initial value "Java is Internet programming language". Print each word of this object in new line.

3. Create a class in java named **StringDemo3**. Create 1 String object with initial value “MALAYALAM”. Check whether string is palindrome or not. Print appropriate message.
4. Create a class in java named **StringDemo4**. Declare 1 string object with initial value “CHRIST”. Print following pattern using charAt( ) of string class.

```
C
CH
CHR
CHRI
CHRIS
CHRIST
```

5. Create a class in java named **StringDemo5**. Declare 1 string object with initial value “CHRIST”. Print following pattern using substring( ) of string class.

```
C
CH
CHR
CHRI
CHRIS
CHRIST
```

6. Create a class in java named **StringDemo5**. Declare a character array with initial value “CHRIST” . Print following pattern using new String(start, noofchar) method of string class.

```
    C
   CH
  CHR
 CHRI
CHRIS
CHRIST
```

7. Create a class in java named **StringDemo6**. Declare 2 string object and check for equality of string and compare one string with the other.

Command line arguments, Arrays and wrapper classes.
All command-line arguments are passed as strings. You must convert numeric values to their internal forms manually.

8. Create a class in java named **CLDemo1** to catch command line arguments and print number of arguments and each argument in new line.

9. Create a class in java named **CLDemo2** to catch command line arguments and print all in uppercase and lowercase.

10. Create a class in java named **CLDemo3** to catch names of cities from command line and store that in string array, sort them by ignoring case and print them.

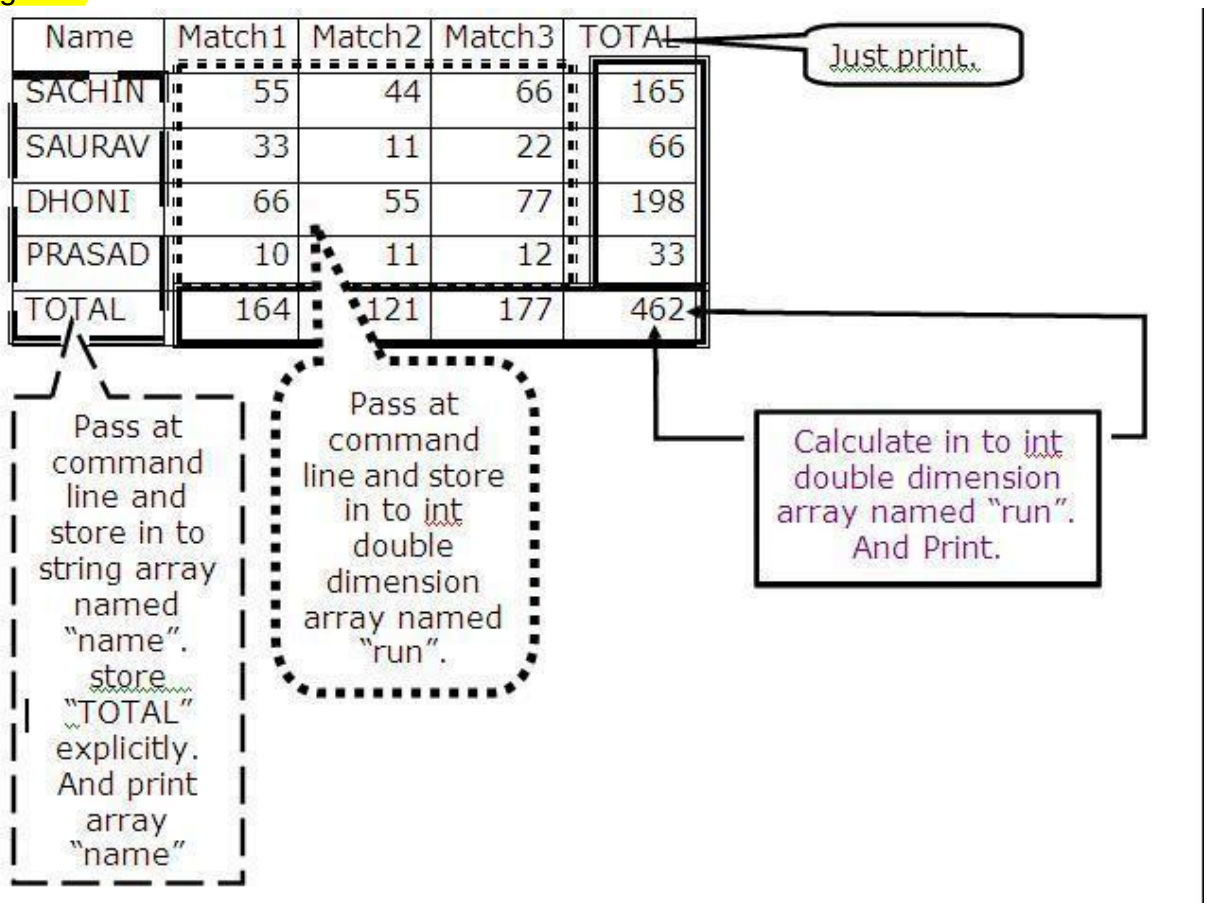
11. Create a class in java named **CLDemo4** which catches names of 10 cities. Find city "RAJKOT" in array. If found, Print index of the city found with city name; otherwise print Message "City was not found".

12. Create a class in java named **CLDemo5** to find the sum of command line arguments and count the invalid integers entered.

13. Create a class in java named **CalcDemo1** which catches two values and sign(+, -, *, /, %) as command line arguments. Perform calculation of two values on the base of sign and print end result.

14. Create a class in java named MeritList to catch names of 10 students and their marks. Sort them on the base of marks. Print them.

15. Create a java class named **CricketDemo** to catch name along with runs of three matches for n cricketers from command line. Store names of all cricketers in to string array named "name". Store runs in to int double-dim array. Make horizontal & vertical sum of runs for each cricketer & for each match. And finally display output as given.



16. Create a class in java DateExample for accepting date and perform various functions using Constructor method.

17. Create a class in java MatrixEx. Declare 3 global double dimension int arrays A,B,C having size of 3 x 3. Assign value to A and B. Define methods print() and add() for performing corresponding operations. Write code in main() to call above methods.

18. Create a class in java StaticDemo having static variables and static method callme(). Call this method callme in main method of StaticByName java class.

19. Create a class in java MathPr and call various static methods of Math class from main() method of MathPr. (eg. Pi, sqrt, pow, sin, cos, tan, log, log10. Refer java.lang.Math in javadocs help manual)

Class, field, methods, visibility control, encapsulation, this, overloading

20. Create a class Employee in java containing private members named empid, empname, gender, address, city, email, mobile. Define public method SetData and GetData for each member. Define print methods which prints all members. Create public class EmpDemo to add Employee class.

21. Create class Person containing private fields named name, bloodgroup, gender, city, email, mobile. Define public getter and setter methods for each member. Define print method which prints all fields. Define toString method which returns full person info in string format. eg “Nilesh, B+, male,Mumbai,nilesh@gmail.com,9898090128”. Create public class **PersonDemo1** to test **Person** class.

Constructor, Overloading of constructor, Default constructor and finalize

22. Create class Date containing private fields named date, month and year. Define public get and set methods for each member. Overload setDate() method which catches all 3 fields together. Define print method which prints date. Define toString method which returns full date in string format. eg “15/8/1947”
    Define no argument constructor which stores date 1-1-2000 or system date to fields. Overload constructor which catches 3 arguments date, month, year. Create public class DateDemo2 to test Date class

23. Create class Shape containing private field named red, green, blue. Define public get and set methods for each field. Define print method which prints class members. Define toString method which returns full Shape info in string format. eg “red:54,green:70,blue:57”. Define no argument and parameterized constructor. Create public class **ShapeDemo2** to execute **Shape** class.

Inheritance: overriding, final, protected, super

24. Create a class Shape in java to find area of different shapes using method overriding.
    Eg: for circle , rectangle, triangle, cylinder.

Open Shape.java file. Save it as **ShapeDemo1.java**. Rename Shape class as ShapeDemo1. Create class **Circle** which extends Shape class. Add private fields like radius. Define public get and set methods for each field. Override print method which prints all fields of super class and the sub class. Override toString method which returns full info of parent and child class in string format. eg “red:55,green:88,blue:44,radius:33”.

25. Define no argument and parameterized constructor in Circle class. Write code in ShapeDemo2.main() method to test Circle class

Inheritance: calling super class constructors.

26. Create a class EmpData in java containing private members named empid, empname, gender, address, city, email, mobile, designation, department, joining_date. Define parameterized constructor in EmpData class : EmpData (empid, empname, gender, address, city, email, mobile, designation, department, joining_date). Define print methods which prints all members. Create public class EmpData to create EmpData object

Inheritance: abstract method and abstract class.

27. Create a class SalaryEmp in java to calculate salary of different department using abstract class.

Package Demonstration and Thread priorities

28. Create a class ThreadPriority in java to implement thread and thread priorities.

29. Create a class NegativeExeption in java to Perform Sorting and to perform Negative index exception handling.

30. Create a class StudApplet in java to accept student details and display the accepted details using Applet.

31. Create a class KeyboardEvent in java for demonstrating some keyboard events

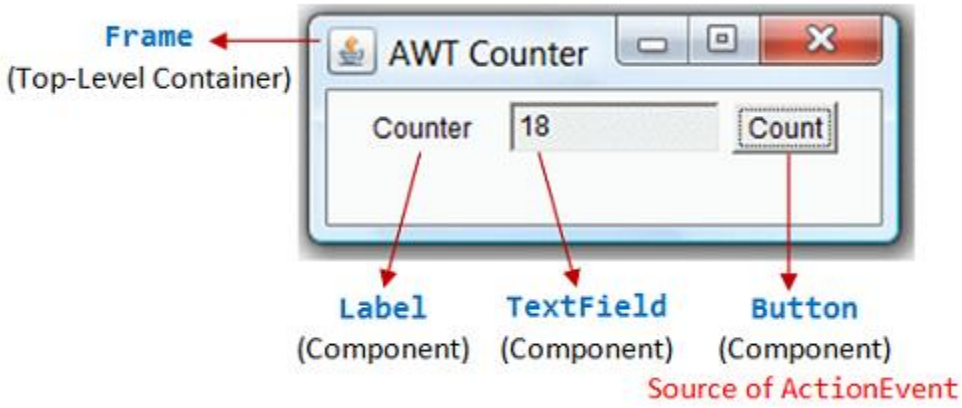
32. Create a class BallMove in java Program to perform movement of a ball inside the applet window

AWT GUI Applications / Applets

33. Write an AWT GUI application (called AWTCounter) as shown in the Figure. Each time the "Count" button is clicked, the counter value shall increase by 1. The program has three components:

    1. a Label "Counter";
    2. a non-editable TextField to display the counter value; and
    3. a Button "Count".

The components are placed inside a container Frame, arranged in FlowLayout.




```
import java.awt.*;         // Using AWT containers and components
import java.awt.event.*;   // Using AWT events and listener interfaces

// An AWT GUI program inherits the top-level container Frame
public class AWTCounter extends Frame implements ActionListener {
   private Label lblCount;     // declare component Label
   private TextField tfCount;  // declare component TextField
   private Button btnCount;    // declare component Button
   private int count = 0;      // counter's value

   // Constructor to setup UI components
   public AWTCounter () {
      setLayout(new FlowLayout());
         // "this" Frame sets layout to FlowLayout, which arranges
         //  Components from left-to-right, then top-to-bottom.

      lblCount = new Label("Counter"); // allocate Label instance
      add(lblCount);                   // "this" Frame adds Label

      tfCount = new TextField(count + "", 10); // allocate
      tfCount.setEditable(false);       // read-only
      add(tfCount);                     // "this" Frame adds tfCount

      btnCount = new Button("Count");   // allocate Button instance
      add(btnCount);                    // "this" Frame adds btnCount
      btnCount.addActionListener(this);
         // btnCount is a source that fires ActionEvent when clicked.
         // The source add "this" object as a listener, which provides
         //  the ActionEvent handler called actionPerformed().
         // Clicking btnCount invokes actionPerformed().

      setSize(250, 100);       // "this" Frame sets initial size
```

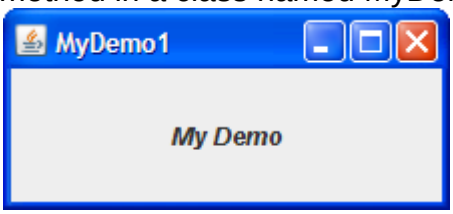
```
    setTitle("AWT Counter"); // "this" Frame sets title
    setVisible(true);        // show "this" Frame
  }

  // ActionEvent handler - Called back when the button has been clicked.
  @Override
  public void actionPerformed(ActionEvent evt) {
    ++count;                 // incrase the counter value
    tfCount.setText(count + ""); // display on the TextField
                             // setText() takes a String
  }

  // The entry main() method
  public static void main(String[] args) {
    // Invoke the constructor by allocating an anonymous instance
    new AWTCounter();
  }
}
```

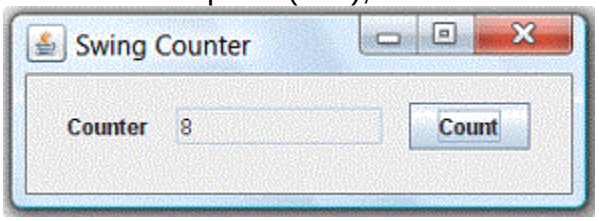
Swing GUI Application

34. Implement a program with a GUI that looks like the one shown below. Put the main method in a class named MyDemo1.



35. Convert the previous AWT exercises AWTCounter to Swing applications called SwingCounter.

    1. Swing Components are kept in package javax.swing. They begin with a prefix "J", e.g., JButton, JLabel, JFrame.
    2. Swing Components are to be added onto the ContentPane of the top-level container JFrame. You can retrieve the ContentPane via method getContentPane() from a JFrame.
    3. Container cp = getContentPane(); // of JFrame
       cp.setLayout(.....);
       cp.add(.....);



For example, SwingCounter.java:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;  // Using Swing components and containers

// A Swing application extends javax.swing.JFrame
public class SwingCounter extends JFrame {
   private JTextField tfCount;
         // Use Swing's JTextField instead of AWT's TextField
   private int count = 0;

   public SwingCounter () {
      // Get the content pane of top-level container Jframe
      // Components are added onto content pane
      Container cp = getContentPane();
      cp.setLayout(new FlowLayout());

      cp.add(new JLabel("Counter"));
      tfCount = new JTextField(count + "", 10);
      tfCount.setEditable(false);
      tfCount.setHorizontalAlignment(JTextField.RIGHT);
      cp.add(tfCount);

      JButton btnCount = new JButton("Count");
      cp.add(btnCount);
      btnCount.addActionListener(new ActionListener() {
        @Override
         public void actionPerformed(ActionEvent e) {
            ++count;
            tfCount.setText(count + "");
         }
      });

      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         // Exit program if Jframe's close-window button clicked
      setSize(300, 100);
      setTitle("Swing Counter");
      setVisible(true);    // show it
   }

   public static void main(String[] args) {
      // Recommended to run the GUI construction in
      //  Event Dispatching thread for thread-safet operations
      SwingUtilities.invokeLater(new Runnable() {
         @Override
```

```
        public void run() {
          new SwingCounter(); // Let the constructor does the job
        }
      });
   }
}
```