

1. Write a Java program to print Fibonacci series using for loop.

```
class Fibonacci
{
    public static void main(String[] args)
    {
        int n = 10,t1=0,t2=1,t3;
        System.out.println("Fibonacci Series are :");
        for (int i = 0; i < n; i++)
        {
            System.out.print(t1+ " ");
            t3 = t1 + t2;
            t1=t2;
            t2=t3;
        }
    }
}
```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac Fibonacci.java
```

```
E:\JAVA\ANU\Anitha\BSC LAB>java Fibonacci
Fibonacci Series are :
0 1 1 2 3 5 8 13 21 34
```

2. Write a Java program to calculate multiplication of 2 matrices.

```
class Matrix
{
    public static void main(String[] args)
    {
        int[][] a = {{1, 2, 3},{4, 5, 6},{7, 8, 9}};
        int[][] b = {{9, 8, 7},{6, 5, 4},{3, 2, 1}};
        int[][] c = new int[3][3];
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                for (int k = 0; k < 3; k++)
                {
                    c[i][j] += a[i][k] * b[k][j];
                }
            }
        }
        System.out.println("Resultant Matrix:");
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                System.out.print(c[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac Matrix.java
```

```
E:\JAVA\ANU\Anitha\BSC LAB>java Matrix
Resultant Matrix:
30 24 18
84 69 54
138 114 90
```

3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.

```

class Rectangle
{
    float length;
    float width;
    void read()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the length : ");
        length = scanner.nextFloat();
        System.out.print("Enter the width : ");
        width = scanner.nextFloat();
    }
    float perimeter()
    {
        return (2 * (length + width));
    }
    float area()
    {
        return (length * width);
    }
    void display()
    {
        System.out.println("Area: " + area());
        System.out.println("Perimeter: " + perimeter());
    }
    public static void main(String args[])
    {
        Rectangle r = new Rectangle();
        r.read();
        r.display();
    }
}

```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac Rectangle.java
```

```

E:\JAVA\ANU\Anitha\BSC LAB>java Rectangle
Enter the length : 6
Enter the width : 4
Area: 24.0
Perimeter: 20.0

```

4. Write a Java program that implements method overloading.

```

class OverLoad
{
    void add(int a,int b)
    {
        System.out.println(a+b);
    }
    void add(float a,float b)
    {
        System.out.println(a+b);
    }
    void add(int a,int b,int c)
    {
        System.out.println(a+b+c);
    }
    public static void main(String args[])
    {
        OverLoad ol=new OverLoad();
        ol.add(10,20);
        ol.add(2.3f,4.4f);
        ol.add(4,2,6);
    }
}

```

```
E:\JAVA\ANU>javac OverLoad.java
```

```

E:\JAVA\ANU>java OverLoad
30
6.7
12

```

5. Write a Java program for sorting a given list of names in ascending order.

```
import java.util.Arrays;
class NameSort
{
    public static void main(String[] args)
    {
        String a[]={"php","java","golang","html","css","python","cpp"};
        Arrays.sort(a);
        System.out.println("Names in ascending order:");
        for (String ele : a)
            System.out.println(ele);
    }
}
```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac NameSort.java
E:\JAVA\ANU\Anitha\BSC LAB>java NameSort
Names in ascending order:
cpp
css
golang
html
java
php
python
```

6. Write a Java program that displays the number of characters, lines and words in a text file.

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
class File
{
    public static void main(String[] args)
    {
        String filePath = "E:/JAVA/ANU/Anitha/BSC LAB/file.txt";
        int charCount = 0;
        int wordCount = 0;
        int lineCount = 0;
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath)))
        {
            String line;
            while ((line = reader.readLine()) != null)
            {
                lineCount++;
                charCount += line.length();
                String[] words = line.split("\\s+");
                wordCount += words.length;
            }
            System.out.println("Number of characters: " + charCount);
            System.out.println("Number of words: " + wordCount);
            System.out.println("Number of lines: " + lineCount);
        }
        catch (IOException e)
        {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        }
    }
}
```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac File.java
E:\JAVA\ANU\Anitha\BSC LAB>java File
Number of characters: 106
Number of words: 19
Number of lines: 3
```

7. Write a Java program to implement various types of inheritance

- i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid

Single Inheritance

```
class Animal
{
    void eat()
    {
        System.out.println("Eating");
    }
}
class Dog extends Animal
{
    void bark()
    {
        System.out.println("Barking");
    }
    public static void main(String[] args)
    {
        Dog d = new Dog();
        d.eat();
        d.bark();
    }
}
```

```
E:\JAVA\ANU>javac Dog.java
```

```
E:\JAVA\ANU>java Dog
Eating
Barking
```

Multi-Level Inheritance

```
class Animal
{
    void eat()
    {
        System.out.println("Eating");
    }
}
class Dog extends Animal
{
    void bark()
    {
        System.out.println("Barking");
    }
}
class BabyDog extends Dog
{
    void sleep()
    {
        System.out.println("Sleeping");
    }
    public static void main(String[] args)
    {
        BabyDog b = new BabyDog();
        b.eat();
        b.bark();
        b.sleep();
    }
}
```

```
E:\JAVA\ANU>javac BabyDog.java
```

```
E:\JAVA\ANU>java BabyDog
Eating
Barking
Sleeping
```

Hierarchical Inheritance

```

class Animal
{
    void eat()
    {
        System.out.println("Eating");
    }
}
class Dog extends Animal
{
    void bark()
    {
        System.out.println("Barking");
    }
}
class Cat extends Animal
{
    void meow()
    {
        System.out.println("Meowing");
    }
    public static void main(String[] args)
    {
        Cat c = new Cat();
        c.eat();
        //c.bark(); ERROR
        c.meow();
    }
}

```

```
E:\JAVA\ANU>javac Cat.java
```

```
E:\JAVA\ANU>java Cat
Eating
Meowing
```

Hybrid Inheritance

```

class Animal
{
    void eat()
    {
        System.out.println("Eating");
    }
}
class Dog extends Animal
{
    void bark()
    {
        System.out.println("Barking");
    }
}
class Cat extends Animal
{
    void meow()
    {
        System.out.println("meowing");
    }
}
class Puppy extends Dog
{
    void sleep()
    {
        System.out.println("Sleeping");
    }
    public static void main(String[] args)
    {
        Puppy b = new Puppy();
        b.eat();
        b.bark();
        b.sleep();
        Cat c=new Cat();
        c.meow();
    }
}

```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac Puppy.java
```

```
E:\JAVA\ANU\Anitha\BSC LAB>java Puppy
Eating
Barking
Sleeping
meowing
```

8. Write a java program to implement runtime polymorphism

```
class Parent
{
    void msg()
    {
        System.out.println("parent Overridden method");
    }
}
class OverRide extends Parent
{
    void msg()
    {
        System.out.println("child Overriding method ");
    }
    public static void main(String args[])
    {
        OverRide or=new OverRide();
        or.msg();
    }
}
```

```
E:\JAVA\ANU>javac OverRide.java
```

```
E:\JAVA\ANU>java OverRide
child Overriding method
```

9. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.

```
import java.util.Scanner;
class InsufficientFundsException extends Exception
{
    InsufficientFundsException(String message)
    {
        super(message);
    }
}
class Atm
{
    float balance;
    Atm(float initialBalance)
    {
        this.balance = initialBalance;
    }
    void withdraw(float amount) throws InsufficientFundsException
    {
        if (amount > balance)
        {
            throw new InsufficientFundsException("Sorry!..Insufficient balance: " + balance);
        }
        balance -= amount;
        System.out.println("Withdrawal successful! Remaining balance: " + balance);
    }
    public static void main(String args[])
    {
        Atm a = new Atm(500f);
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the amount to withdraw: ");
        float amount = scanner.nextFloat();
        try
        {
            a.withdraw(amount);
        }
        catch (InsufficientFundsException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

```
E:\JAVA\ANU\Anitha\BSC LAB>javac Atm.java
```

```
E:\JAVA\ANU\Anitha\BSC LAB>java Atm
Enter the amount to withdraw: 300
Withdrawal successful! Remaining balance: 200.0
```

```
E:\JAVA\ANU\Anitha\BSC LAB>java Atm
Enter the amount to withdraw: 700
Sorry!..Insufficient balance: 500.0
```

10. Write a Java program to create three threads and that displays “good morning”, for every one second, “hello” for every 2 seconds and “welcome” for every 3 seconds by using extending Thread class.

```
class MorningThread extends Thread
{
    public void run()
    {
        try
        {
            while (true)
            {
                System.out.println("Good Morning");
                Thread.sleep(1000);
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
class HelloThread extends Thread
{
    public void run()
    {
        try
        {
            while (true)
            {
                System.out.println("Hello");
                Thread.sleep(2000);
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
class WelcomeThread extends Thread
{
    public void run()
    {
        try
        {
            while (true)
            {
                System.out.println("Welcome");
                Thread.sleep(3000);
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
class MultiThread
{
    public static void main(String[] args)
    {
        MorningThread morning = new MorningThread();
        HelloThread hello = new HelloThread();
        WelcomeThread welcome = new WelcomeThread();
        morning.start();
        hello.start();
        welcome.start();
    }
}
```

```

E:\JAVA\ANU\Anitha\BSC LAB>javac MultiThread.java

E:\JAVA\ANU\Anitha\BSC LAB>java MultiThread
Welcome
Good Morning
Hello
Good Morning
Hello
Good Morning
Welcome
Good Morning
Hello
Good Morning
Good Morning
Welcome
Hello
Good Morning
Good Morning
Hello

```

11. Write a Java program that creates three threads. First thread displays “OOPS”, the second thread displays “Through” and the third thread Displays “JAVA” by using Runnable interface.

```

class Messages implements Runnable
{
    private String message;
    Messages(String message)
    {
        this.message = message;
    }
    public void run()
    {
        System.out.println(message);
    }
    public static void main(String args[])
    {
        Messages m1=new Messages("OOPS");
        Thread t1 = new Thread(m1);
        Messages m2=new Messages("Through");
        Thread t2 = new Thread(m2);
        Messages m3=new Messages("JAVA");
        Thread t3 = new Thread(m3);
        t1.start();
        t2.start();
        t3.start();
    }
}

```

```

E:\JAVA\ANU\Anitha\BSC LAB>javac Messages.java

E:\JAVA\ANU\Anitha\BSC LAB>java Messages
OOPS
Through
JAVA

```



```

import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
public class MouseEventDemo
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("Mouse Event Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(600, 600);
        JPanel panel = new JPanel()
        {
            @Override
            protected void paintComponent(Graphics g)
            {
                super.paintComponent(g);
                g.drawString("Mouse Event Demo", 150, 150);
            }
        };
        panel.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseEntered(MouseEvent e)
            {
                System.out.println("Mouse entered the panel");
            }
            @Override
            public void mouseExited(MouseEvent e)
            {
                System.out.println("Mouse exited the panel");
            }
            @Override
            public void mouseClicked(MouseEvent e)
            {
                System.out.println("Mouse clicked at: " + e.getPoint());
            }
            @Override
            public void mousePressed(MouseEvent e)
            {
                System.out.println("Mouse pressed at: " + e.getPoint());
            }
            @Override
            public void mouseReleased(MouseEvent e)
            {
                System.out.println("Mouse released at: " + e.getPoint());
            }
        });
        panel.addMouseMotionListener(new MouseAdapter()
        {
            @Override
            public void mouseDragged(MouseEvent e)
            {
                System.out.println("Mouse dragged at: " + e.getPoint());
            }
            @Override
            public void mouseMoved(MouseEvent e)
            {
                System.out.println("Mouse moved at: " + e.getPoint());
            }
        });
        frame.add(panel);
        frame.setVisible(true);
    }
}

```

E:\JAVA\ANU\Anitha\BSC LAB>javac MouseEventDemo.java

E:\JAVA\ANU\Anitha\BSC LAB>java MouseEventDemo

Mouse entered the panel

Mouse moved at: java.awt.Point[x=551,y=456]

Mouse pressed at: java.awt.Point[x=551,y=456]

Mouse released at: java.awt.Point[x=551,y=456]

Mouse clicked at: java.awt.Point[x=551,y=456]

Mouse moved at: java.awt.Point[x=551,y=456]

Mouse dragged at: java.awt.Point[x=521,y=413]

Mouse released at: java.awt.Point[x=521,y=413]

Mouse moved at: java.awt.Point[x=540,y=405]

Mouse moved at: java.awt.Point[x=578,y=394]

Mouse exited the panel



