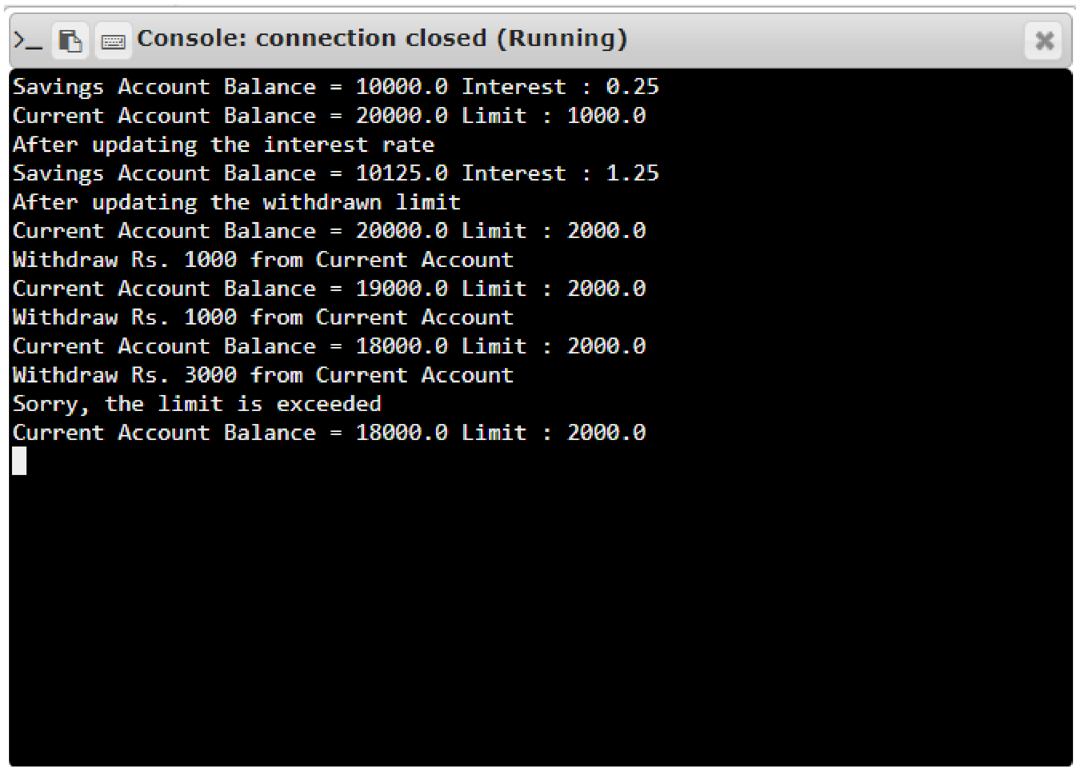
PROGRAM 4.1

```
class Account

{

   private double bal;

   //private int accnum

   public Account (double bl)

   {

      bal=bl;

   }

   public void deposit(double sum)

   {

      if (sum>0)

         bal+=sum;

      else

         System.err.println("Account.deposit(...):"+"cannot deposit negative amount.");


   }

   public void withdraw(double sum)

   {

      if (sum>0)

         bal-=sum;

      else

         System.err.println("Account.withdraw(...):"+"cannot withdraw negative amount.");

   }
```

```
    public double getBalance()
    {
        return bal;
    }
    public final void intprint(double in)
    {
        System.out.println("Savings Account Balance = "+bal+" Interest : "+in);
    }
    public final void limitprint(double l)
    {
        System.out.println("Current Account Balance = "+bal+" Limit : "+l);
    }
}
class savingsAccount extends Account{
    double interest;
    savingsAccount(double b,double i){
        super(b);
        this.interest=i;
        super.intprint(interest);
    }
    public void updateinterest(double i){
        interest=i;
        System.out.println("After updating the interest rate");
    }
    public void addinterest(double i){
        double b1,j;
        b1=super.getBalance();
        j=(b1*i)/100;
        super.deposit(j);
        super.intprint(interest);
    }
}
class currentAccount extends Account{
    double limit;
    currentAccount(double b,double li){
        super(b);
        this.limit=li;
        super.limitprint(limit);
    }
    public void updatelimit(double li){
        limit=li;
        System.out.println("After updating the withdrawn limit");
        super.limitprint(li);
    }
    public void checklimit(double amt){
        if(amt<=limit){
            super.withdraw(amt);
            System.out.println("Withdraw Rs. "+(int)amt+" from Current Account");
            super.limitprint(limit);
        }
        else{
            System.out.println("Withdraw Rs. "+(int)amt+" from Current Account");
            System.out.println("Sorry, the limit is exceeded");
            super.limitprint(limit);
        }
    }
}
public class exercise3{
    public static void main(String args[]){
        savingsAccount ac=new savingsAccount(10000,0.25);
        currentAccount acc=new currentAccount(20000.0,1000.0);
        ac.updateinterest(1.25);
        ac.addinterest(1.25);
        acc.updatelimit(2000.0);
        acc.checklimit(1000.0);
        acc.checklimit(1000.0);
        acc.checklimit(3000.0);
    }
}
```

OUTPUT



```
Savings Account Balance = 10000.0 Interest : 0.25
Current Account Balance = 20000.0 Limit : 1000.0
After updating the interest rate
Savings Account Balance = 10125.0 Interest : 1.25
After updating the withdrawn limit
Current Account Balance = 20000.0 Limit : 2000.0
Withdraw Rs. 1000 from Current Account
Current Account Balance = 19000.0 Limit : 2000.0
Withdraw Rs. 1000 from Current Account
Current Account Balance = 18000.0 Limit : 2000.0
Withdraw Rs. 3000 from Current Account
Sorry, the limit is exceeded
Current Account Balance = 18000.0 Limit : 2000.0
```

PROGRAM 4.2

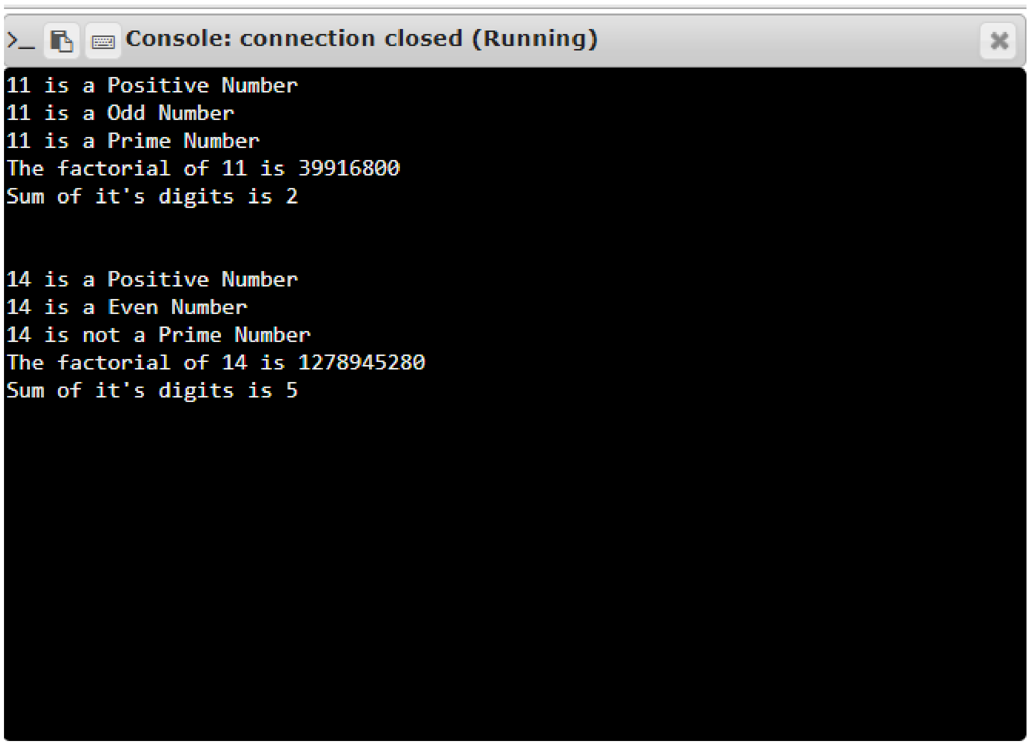
```
interface IntOperations {
        void integer();
        void prime();
        void evenOdd();
        void factorial();
        void Sumofdigit();
}
class MyNumber implements IntOperations {
        int i = 0 , no;
        MyNumber() {
                no = 0;
        }
        MyNumber(int num) {
                no = num;
        }
        public void integer() {

                if (no < 0) {
                        System.out.println(no + " is a Negative Number");
                } else if (no > 0) {
                        System.out.println(no + " is a Positive Number");
                } else {
                        System.out.println(no + " is a Positive Number");
```

```
		}


	}
	public void prime() {
		int flag = 0;
		for (i = 2; i < no; i++) {
			if (no % i == 0) {
				flag = 1;
			}
		}

		if (flag == 1) {
			System.out.println(no + " is not a Prime Number");
		} else {
			System.out.println(no + " is a Prime Number");
		}

	}
	public void evenOdd() {

		if (no % 2 == 0) {
			System.out.println(no + " is a Even Number");
		} else {
			System.out.println(no + " is a Odd Number");
```

```
        }

    }
    public void factorial(){
        int fact = 1;
        for (i = 1; i <= no; i++) {
            fact = fact * i;
        }

        System.out.println("The factorial of " + no + " is " + fact);

    }
    public void Sumofdigit() {
        int sum = 0, rem = 0, n = 0;
        while (no > 0) {
            rem = no % 10;
            sum = sum + rem;
            no = no / 10;
        }

        System.out.println("Sum of it's digits is " + sum);

    }
}
```

```
class exercise4{

        public static void main(String  args[]){

                        MyNumber m = new MyNumber(11);

                        m.integer();

                        m.evenOdd();

                        m.prime();

                        m.factorial();

                        m.Sumofdigit();

                        System.out.println("\n");

                        MyNumber n = new MyNumber(14);

                        n.integer();

                        n.evenOdd();

                        n.prime();

                        n.factorial();

                        n.Sumofdigit();

        }

}
```

OUTPUT



```
11 is a Positive Number
11 is a Odd Number
11 is a Prime Number
The factorial of 11 is 39916800
Sum of it's digits is 2


14 is a Positive Number
14 is a Even Number
14 is not a Prime Number
The factorial of 14 is 1278945280
Sum of it's digits is 5
```

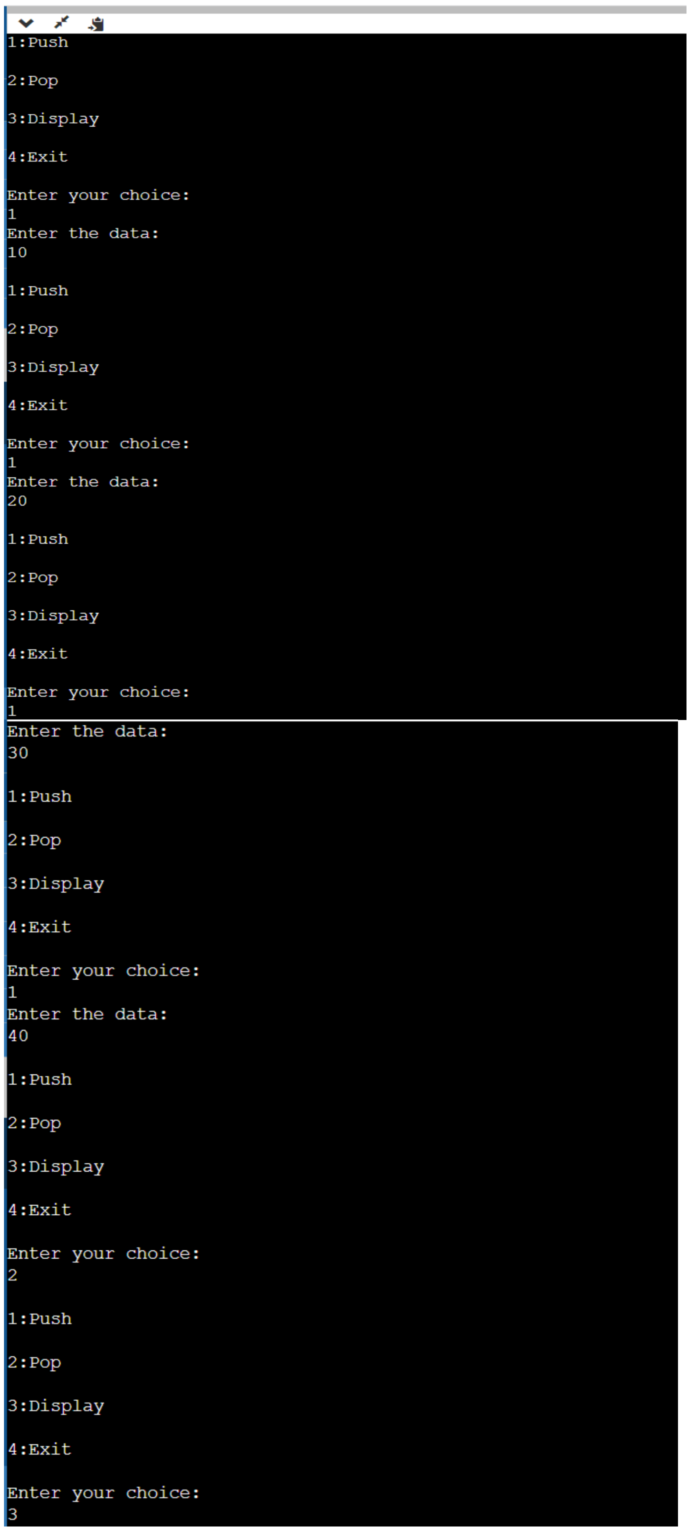
PROGRAM 4.3

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

interface StackOperations{

   int max = 5;

   void push(int data);

   void pop();

   int isempty();

   int isfull();

}

class MyStack implements StackOperations{

   private int arr[]=new int[StackOperations.max];

   private int top;

   public MyStack(){top = 0;}

   public void push(int data){arr[top++]=data;}

   public void pop(){top--;}

   public int isempty(){

      if(top==0){return 1;}

      else{return 0;}

   }

   public int isfull(){

      if (top == max){return 1;}

      else{return 0;}

   }
```

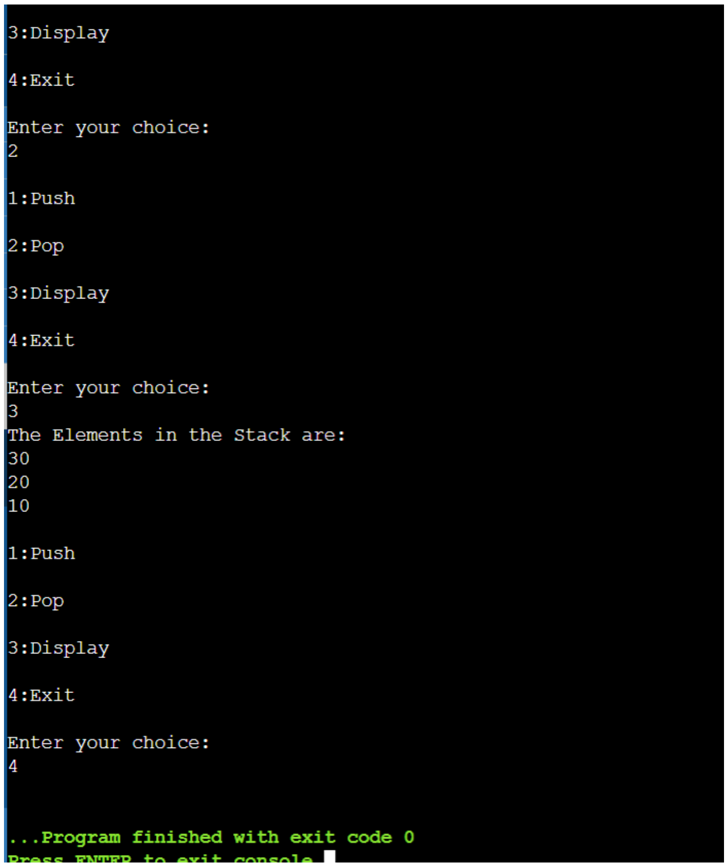
```
    public void display(){
        int i;
        if(top>0){
            System.out.println("The Elements in the Stack are: ");
            for(i=top-1;i>=0;i--)
                System.out.println(arr[i]);
        }else{isempty();}
    }
}
public class exercise5{
    public static void main(String[] args) throws Exception{
        int ch, data;
        String c;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        MyStack s = new MyStack();
        do{
            System.out.println("1:Push");
            System.out.println("\n2:Pop");
            System.out.println("\n3:Display");
            System.out.println("\n4:Exit");
            System.out.println("\nEnter your choice:");
            ch = Integer.parseInt(br.readLine());
            switch(ch){
                case 1:
```

```
                if (s.isfull()==1){System.out.println("Stack is full");}
                else{
                    System.out.println("Enter the data:");
                    data = Integer.parseInt(br.readLine());
                    s.push(data);
                }break;
            case 2:
                if(s.isempty()==1){System.out.println("Stack is empty");}
                else{s.pop();}break;
            case 3:
                if (s.isempty()==1){System.out.println("Stack is empty");}
                else{s.display();}break;
            case 4:
                System.exit(0);
            case 5:
                System.out.println("\nInvalid choice");
                break;
        }
        System.out.println();
    }while(ch!=4);
  }
}
```

OUTPUT



```
1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
1
Enter the data:
10

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
1
Enter the data:
20

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
1
Enter the data:
30

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
1
Enter the data:
40

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
2

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
3
```

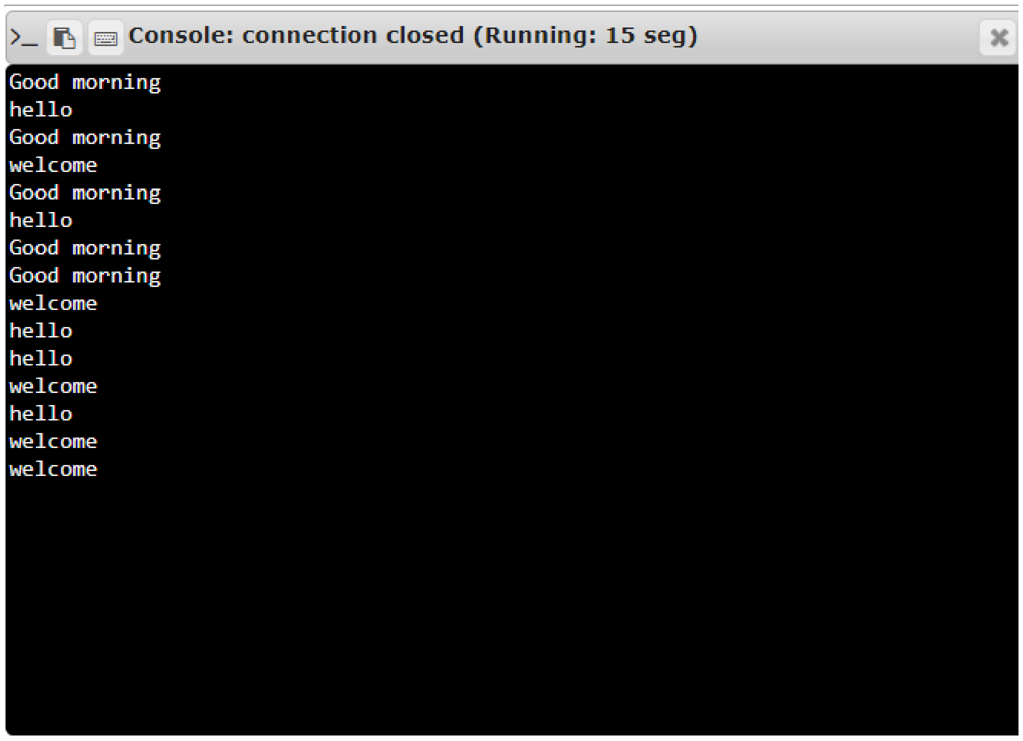


```
3:Display

4:Exit

Enter your choice:
2

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
3
The Elements in the Stack are:
30
20
10

1:Push

2:Pop

3:Display

4:Exit

Enter your choice:
4


...Program finished with exit code 0
Press ENTER to exit console.
```

PROGRAM 5.1

```
class GoodMorning extends Thread {
    synchronized public void run() {
        try {
            int i=0;
            while (i<5) {
                sleep(1000);
                System.out.println("Good morning ");
                i++;
            }
        } catch (Exception e) {
        }
    }
}

class Hello extends Thread {
    synchronized public void run() {
        try {
            int i=0;
            while (i<5) {
                sleep(2000);
                System.out.println("hello");
                i++;
            }
        } catch (Exception e) {

        }

    }

}


class Welcome extends Thread {

    synchronized public void run() {

        try {

            int i=0;

            while (i<5) {

                sleep(3000);

                System.out.println("welcome");

                i++;

            }

        } catch (Exception e) {

        }

    }

}


class MultithreadDemo {

    public static void main(String args[]) {

        GoodMorning t1 = new GoodMorning();

        Hello t2 = new Hello();

        Welcome t3 = new Welcome();
        t1.start();

        t2.start();

        t3.start();


    }

}
```

OUTPUT



Console: connection closed (Running: 15 seg)

```
Good morning
hello
Good morning
welcome
Good morning
hello
Good morning
Good morning
welcome
hello
hello
welcome
hello
welcome
welcome
```

PROGRAM 5.2

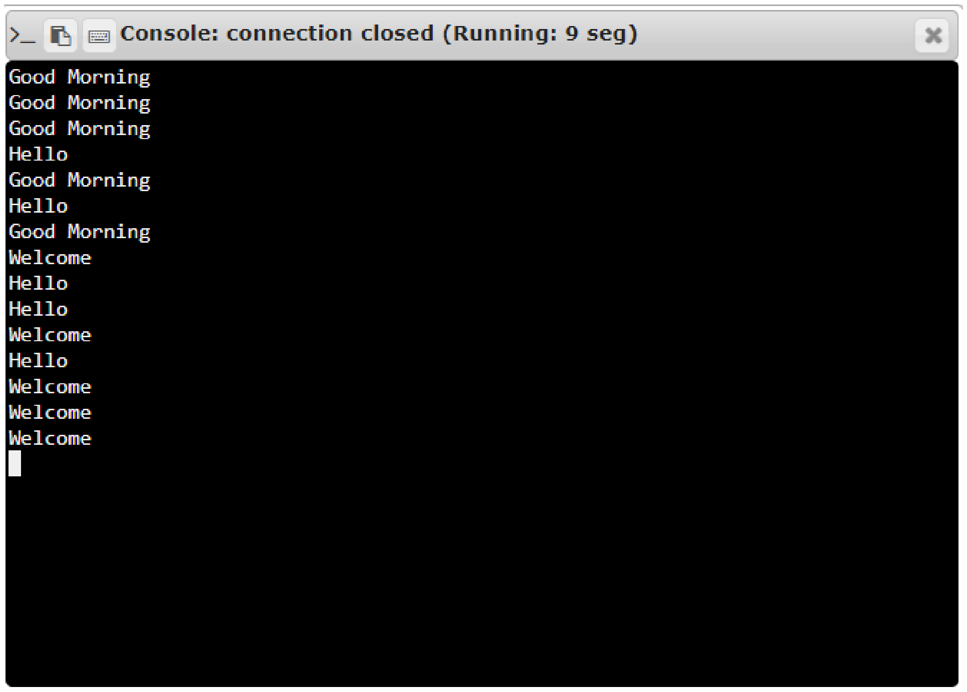
```
}

import java.lang.*;

import java.util.*;

import java.awt.*;

class One implements Runnable

{

One()

{

new Thread(this,"one").start();

try

{

Thread.sleep(1000);

}

catch(InterruptedException e)

{

}

}

public void run()

{

for(int i = 0;i<5;i++)

{

try

{

Thread.sleep(1000);

}
```

```
catch(InterruptedException e)

{

}

System.out.println("Good Morning");

}

}

}

class Two implements Runnable

{

Two()

{

new Thread(this,"two").start();

try

{

Thread.sleep(2000);

}

catch(InterruptedException e)

{

}

}

public void run()

{

for(int j=0;j<5;j++)

{
```

```
try
{
Thread.sleep(2000);
}
catch(InterruptedException e)
{
}
System.out.println("Hello");
}
}
}
class Three implements Runnable
{
Three()
{
new Thread(this,"Three").start();
try
{
Thread.sleep(3000);
}
catch(InterruptedException e)
{
}
}
```

```
public void run()

{

for(int k = 0;k<5;k++)

{

try

{

Thread.sleep(3000);

}

catch(InterruptedException e)

{

}

System.out.println("Welcome");

}

}

}

class MyThread

{

public static void main(String args[])

{

One obj1=new One();

Two obj2=new Two();

Three obj3=new Three();

}

}
```

OUTPUT



```
Good Morning
Good Morning
Good Morning
Hello
Good Morning
Hello
Good Morning
Welcome
Hello
Hello
Welcome
Hello
Welcome
Welcome
Welcome
```