Strings

String constructors

```
chars chars[] = {'b','m','s','c','e'}
String s1 = new String(chars)
String s2 = new String(chars,1,3);
output : bmsce
         msc
```

String length

```
Char chars[] = { 'p','y','t','h','o','n'}
String s= new String(chars);
String.out.println(s.length());
output: 6
```

String literal & Concontetation

```
System.out.println("abc".length);
String car ="BMW";
System.out.println ("He has " +car+ "car");
output : 2
        He has a BMW car
```

getchars

String clg = "Welcome bmsce college";

getchars(11,17,buff,0);

output : Bmsce

equals & equalsIgnore case:

Bmsce equals Bmsce -> true

Bmsce equals College -> false

Bmsce equals BMSCE -> false

Bmsce equalsIgnoreCase BMSCE -> true

region matches

Boolean ismatch = str.region Matches (11,Str2, 0,13);

output : substring is matched.

Startswith and endswith()

String game = "Basketball".

System.out.println (game.startswith ("Basket"));

System.out.println (game. endswith ("ball"));

output:

true

true

equals v/s ==

Hello equals Hello -> true

Hello == Hello -> false

Sort

apple ball cat dog end free gun hen ice jug kite lift man net orange parrot queen ring star umbrella van watch Xmastree yak zee

12)

2 3 4 5 6 7 8 9

13

This is a test . This is, too .

14

World

15

Colllge

16

Hello friends

17

Student1

name: [redacted] Elsaran

Regno: 253

Sem: 3

CGPA: 8.135

Student2

name: suresh

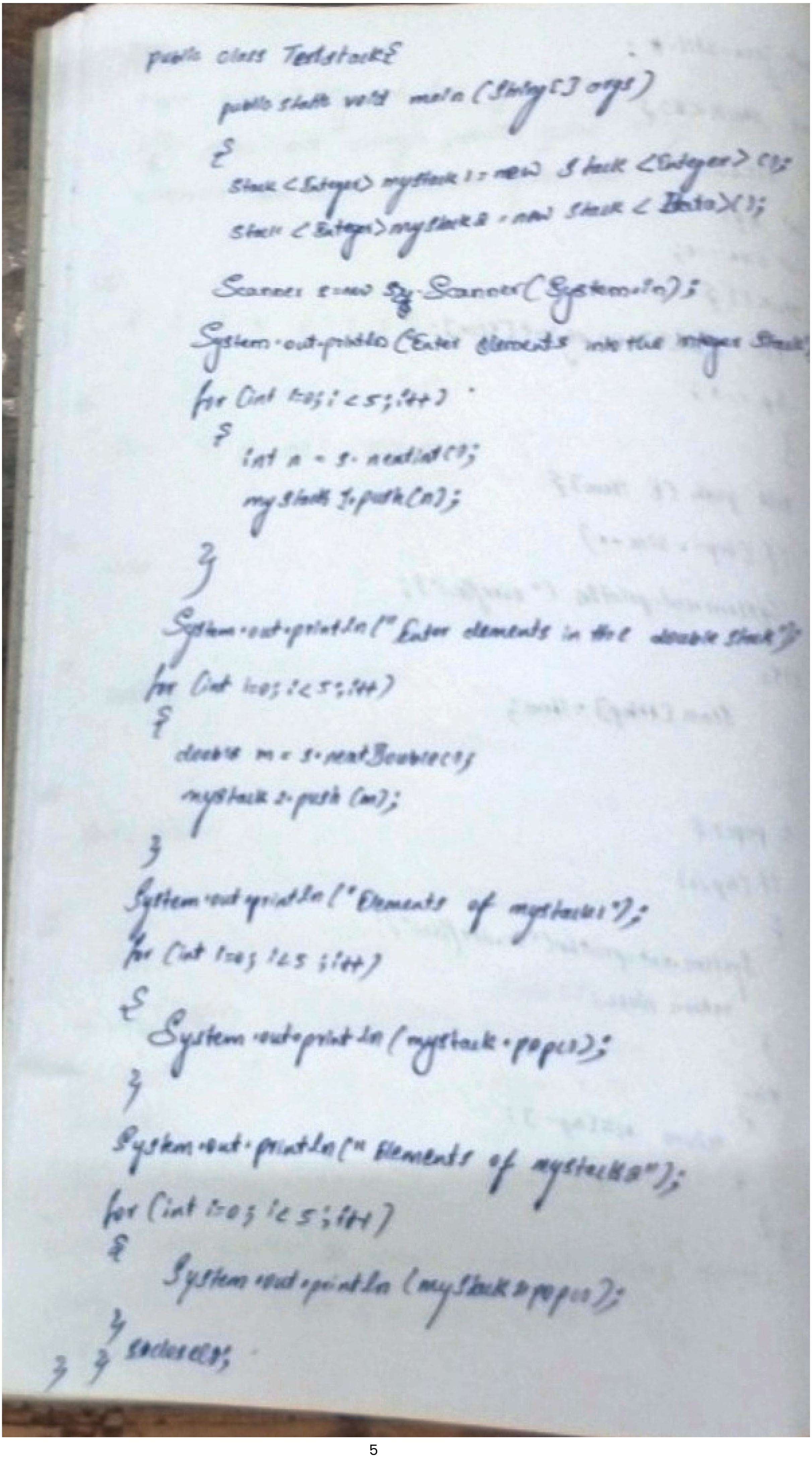
Regno: 230

Sem: 3

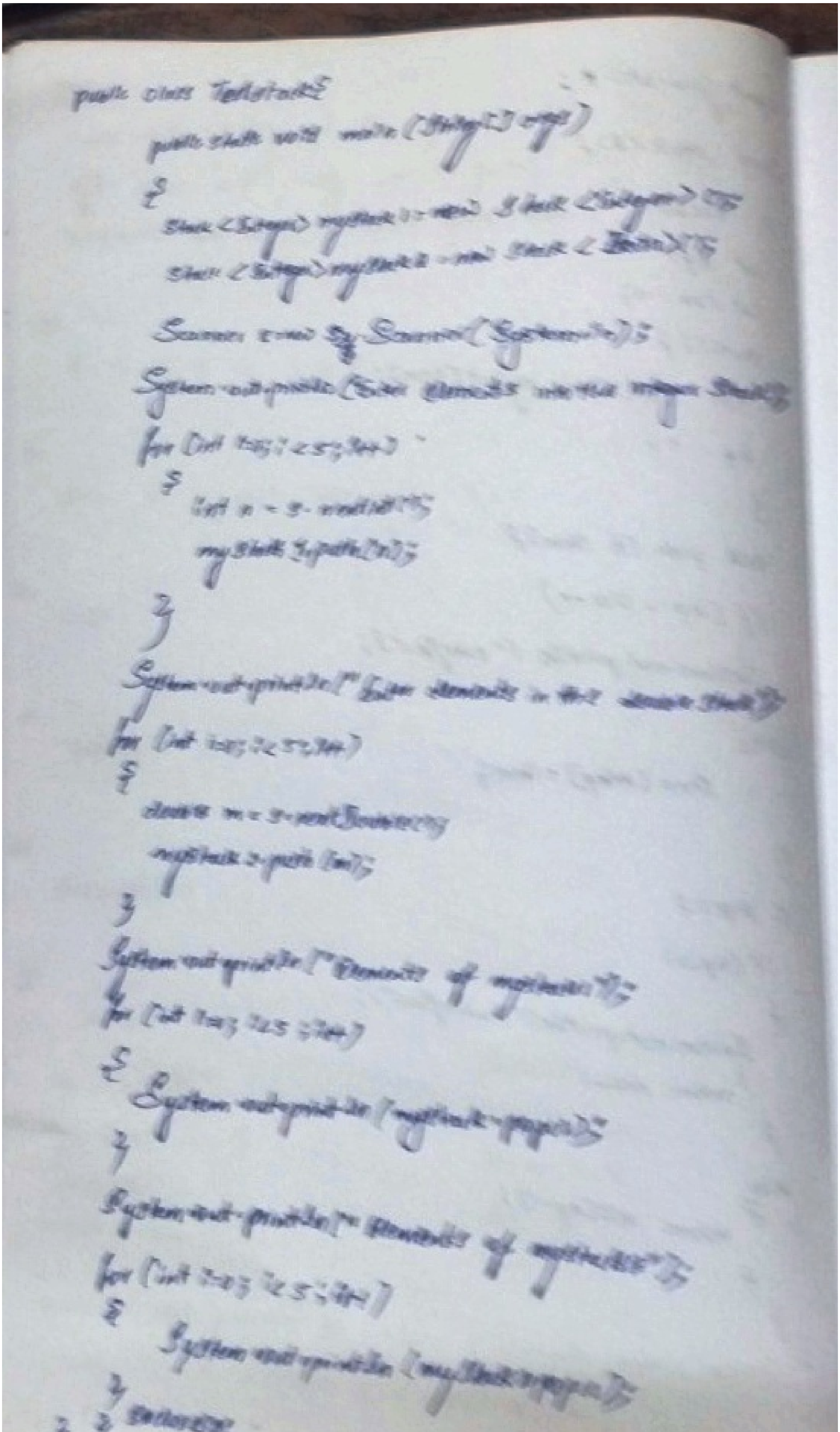
CGPA: 9.5

Write a java program to create a generic class stack which holds 5 integer & 5 double values.

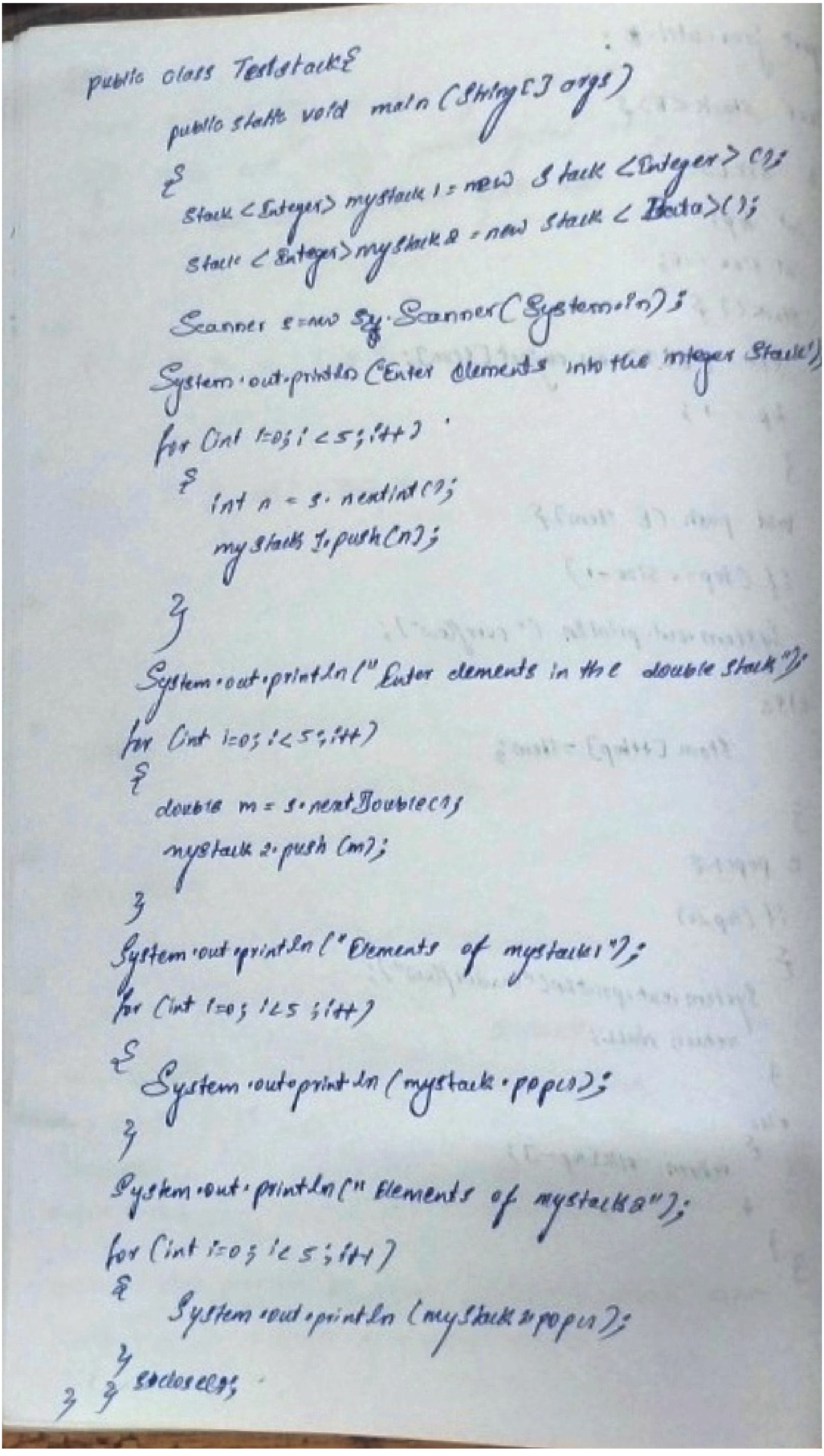
```
Import java.util.*;
Class Stack<E>{
 E stk[];
  int top;
  int size =10;
  Stack(){
   stk = (E[])new object[size];
   top =-1;
  }
  void push (E item){
  if (top== size -1)
  System.out.println ("overflow");
 else
      Stack[++top] = item;
 }
 E pop(){
  if (top<0)
  {
   System.out.println("underflow");
    return NULL;
  }
  else
   {
    return stk[top--];
   }
  }
 }
```



```
public class Teststack{
        public static void main(String[] args)
        {
          Stack <Integer> mystack1 = new Stack <Integer> ();
          Stack <Integer> mystack2 = new Stack < Data>();

        Scanner s=new ~~Sy~~.Scanner(System.in);
      System.out.println("Enter elements into the integer Stack");
      for (int i=0;i<5;i++)
        {
          int n = s.nextInt();
          my stack1.push(n);

      }
     System.out.println("Enter elements in the double stack");
    for (int i=0;i<5;i++)
    {
      double m = s.nextDouble();
      mystack2.push (m);
    }
   System.out.println("Elements of mystack1");
   for (int i=0; i<5 ;i++)
   {
     System.out.println(mystack.pop());
   }
   System.out.println("Elements of mystack2");
  for (int i=0;i<5;i++)
  {
     System.out.println (myStack2.pop());
  }
}  } s.close();
```



```
public class TestStack{
    public static void main(String[] args)
    {
      Stack<Integer> myStack1 = new Stack<Integer>();
      Stack<Integer> myStack2 = new Stack<Double>();

      Scanner s=new Scanner(System.in);
     System.out.println("Enter elements into the integer Stack");
     for (int i=0;i<5;i++)
      {
        int n = s.nextInt();
        myStack1.push(n);
      }
    System.out.println("Enter elements in the double Stack");
   for (int i=0;i<5;i++)
   {
     double m = s.nextDouble();
     myStack2.push(m);
   }
   System.out.println("Elements of myStack1");
   for (int i=0; i<5 ;i++)
   {
     System.out.println(myStack1.pop());
   }
   System.out.println("Elements of myStack2");
   for (int i=0; i<5 ;i++)
   {
       System.out.println(myStack2.pop());
   }
 } } TestStack;
```



```
public class Teststack{
    public static void main(String[] args)
    {
     Stack <Integer> mystack1 = new Stack <Integer>();
     Stack <Integer> mystack2 = new Stack < Double>();

    Scanner s=new ~~Sy~~.Scanner(System.in);
   System.out.println("Enter elements into the integer Stack");
   for (int i=0; i<5; i++)
    {
      int n = s.nextInt();
      my stack 1.push(n);

    }
  System.out.println("Enter elements in the double stack");
 for (int i=0; i<5; i++)
 {
   double m = s.nextDouble();
    mystack 2.push (m);
 }
System.out.println ("Elements of mystack1");
for (int i=0; i<5 ;i++)
{
  System.out.println (mystack.pop());
 }
System.out.println(" Elements of mystack2");
for (int i=0; i<5 ; i++)
{
    System.out.println (myStack2.pop());
 }
 } s.close();
} }
```