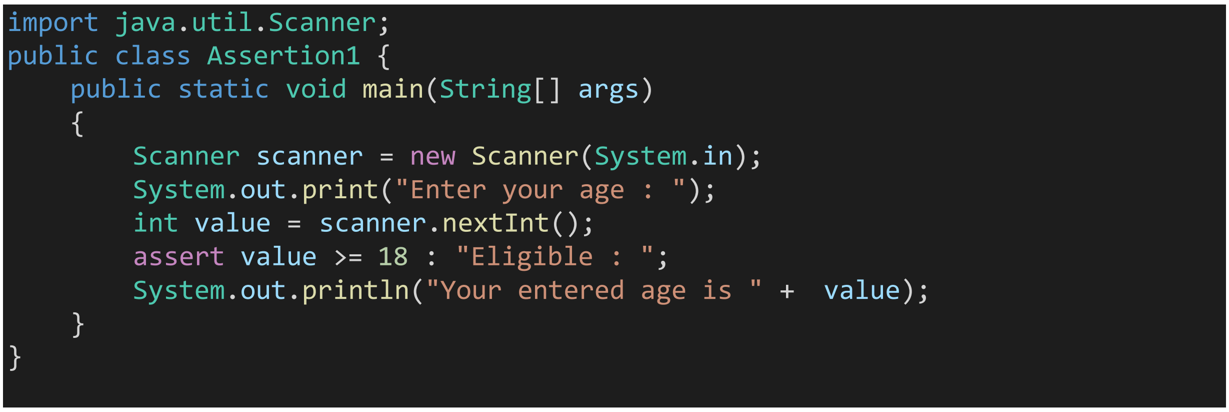
Q. Assertion – 1

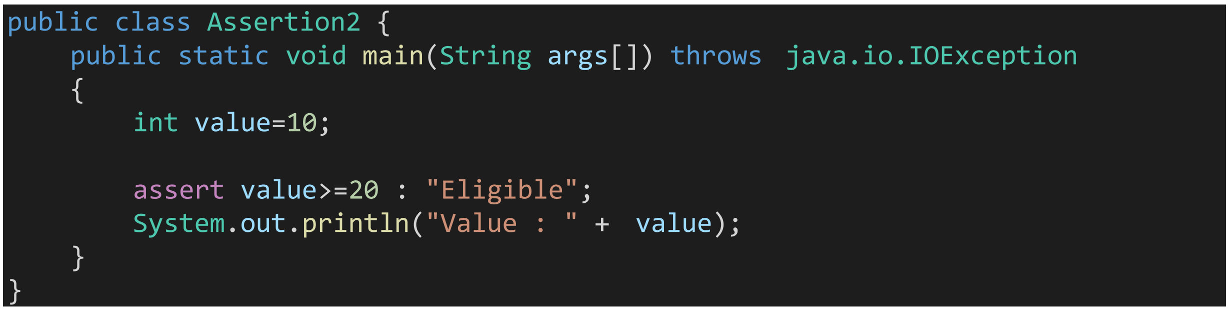


```java
import java.util.Scanner;
public class Assertion1 {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your age : ");
        int value = scanner.nextInt();
        assert value >= 18 : "Eligible : ";
        System.out.println("Your entered age is " +  value);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Assertion1.java

Enter your age : 21

Your entered age is 21
```

Q. Assertion – 2

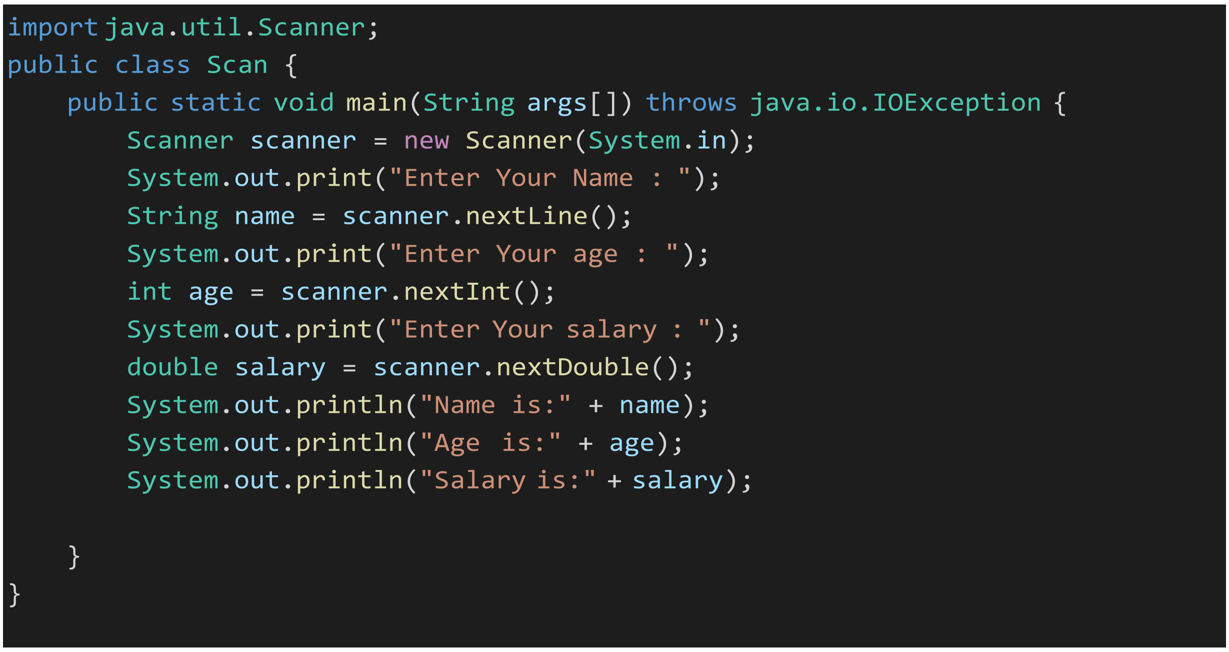


```java
public class Assertion2 {
    public static void main(String args[]) throws  java.io.IOException
    {
        int value=10;

        assert value>=20 : "Eligible";
        System.out.println("Value : " +  value);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Assertion2.java

Value : 10
```

Q. Example of Scanner



```java
import java.util.Scanner;
public class Scan {
    public static void main(String args[]) throws java.io.IOException {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Your Name : ");
        String name = scanner.nextLine();
        System.out.print("Enter Your age : ");
        int age = scanner.nextInt();
        System.out.print("Enter Your salary : ");
        double salary = scanner.nextDouble();
        System.out.println("Name is:" + name);
        System.out.println("Age  is:" + age);
        System.out.println("Salary is:" + salary);

    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Scan.java

Enter Your Name : Pradip

Enter Your age : 21

Enter Your salary : 25000

Name is:Pradip

Age is:21

Salary is:25000.0
```

Q. System Properties

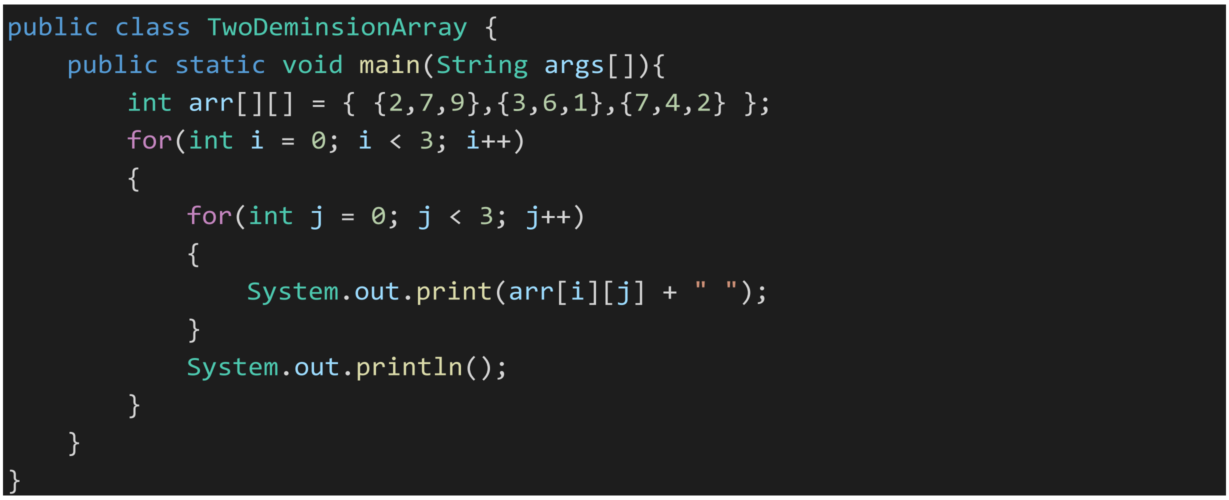
```java
import java.util.*;
public class GetProperties{
    public static void main(String[] args)
    {
        Properties properties = System.getProperties();
        properties.list(System.out);

    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\GetProperties.java

-- listing properties --

java.specification.version=13

sun.cpu.isalist=amd64

sun.jnu.encoding=Cp1252

java.class.path=.

java.vm.vendor=Oracle Corporation

sun.arch.data.model=64

user.variant=

java.vendor.url=https://java.oracle.com/

java.vm.specification.version=13

os.name=Windows 10

sun.java.launcher=SUN_STANDARD

user.country=IN

sun.boot.library.path=C:\Program Files\Java\jdk-13.0.1\bin
sun.java.command=jdk.compiler/com.sun.tools.javac.laun...
jdk.debug=release
sun.cpu.endian=little
user.home=C:\Users\Xtrem
user.language=en
sun.stderr.encoding=cp850
java.specification.vendor=Oracle Corporation
java.version.date=2019-10-15
java.home=C:\Program Files\Java\jdk-13.0.1
file.separator=\
java.vm.compressedOopsMode=32-bit
line.separator=

sun.stdout.encoding=cp850
java.vm.specification.vendor=Oracle Corporation
java.specification.name=Java Platform API Specification
jdk.module.main.class=com.sun.tools.javac.launcher.Main
user.script=
sun.management.compiler=HotSpot 64-Bit Tiered Compilers
java.runtime.version=13.0.1+9
user.name=Xtrem
jdk.launcher.sourcefile=D:\MCA\MCA SEM 3\JAVA\Practice\.\GetP...
```

```
path.separator=;
os.version=10.0
java.runtime.name=Java(TM) SE Runtime Environment
file.encoding=Cp1252
java.vm.name=Java HotSpot(TM) 64-Bit Server VM
java.vendor.url.bug=https://bugreport.java.com/bugreport/
java.io.tmpdir=C:\Users\Xtrem\AppData\Local\Temp\
java.version=13.0.1
user.dir=D:\MCA\MCA SEM 3\JAVA\Practice
os.arch=amd64
java.vm.specification.name=Java Virtual Machine Specification
sun.os.patch.level=
java.library.path=C:\Program Files\Java\jdk-13.0.1\bin;...
java.vm.info=mixed mode, sharing
java.vendor=Oracle Corporation
java.vm.version=13.0.1+9
sun.io.unicode.encoding=UnicodeLittle
java.class.version=57.0
```

Q. 2D Array

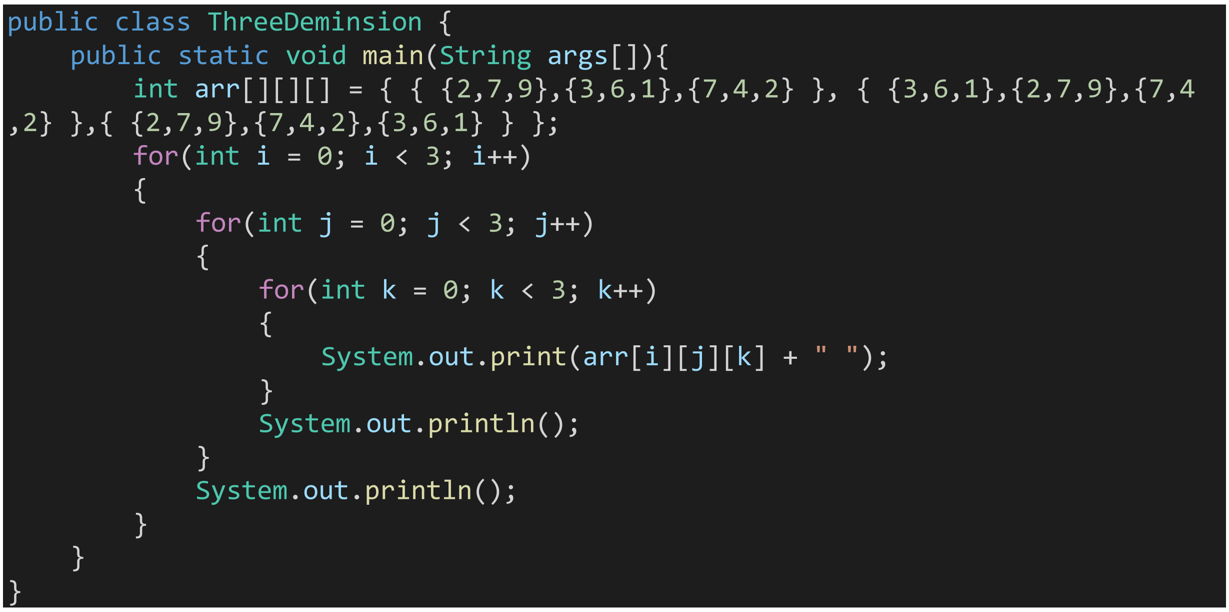


```java
public class TwoDeminsionArray {
    public static void main(String args[]){
        int arr[][] = { {2,7,9},{3,6,1},{7,4,2} };
        for(int i = 0; i < 3; i++)
        {
            for(int j = 0; j < 3; j++)
            {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\TwoDeminsionArray.java

2 7 9

3 6 1

7 4 2
```

Q. 3D Array

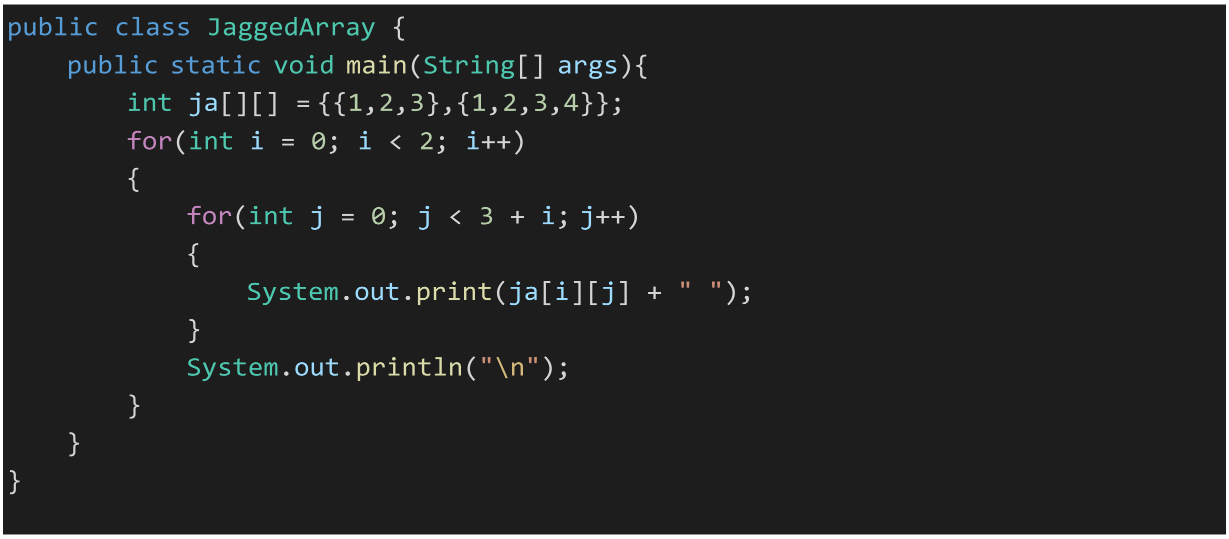


```java
public class ThreeDeminsion {
    public static void main(String args[]){
        int arr[][][] = { { {2,7,9},{3,6,1},{7,4,2} }, { {3,6,1},{2,7,9},{7,4
,2} },{ {2,7,9},{7,4,2},{3,6,1} } };
        for(int i = 0; i < 3; i++)
        {
            for(int j = 0; j < 3; j++)
            {
                for(int k = 0; k < 3; k++)
                {
                    System.out.print(arr[i][j][k] + " ");
                }
                System.out.println();
            }
            System.out.println();
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\ThreeDeminsion.java

2 7 9

3 6 1

7 4 2



3 6 1

2 7 9

7 4 2



2 7 9

7 4 2

3 6 1
```

Q. Jagged Array



```java
public class JaggedArray {
    public static void main(String[] args){
        int ja[][] ={{1,2,3},{1,2,3,4}};
        for(int i = 0; i < 2; i++)
        {
            for(int j = 0; j < 3 + i; j++)
            {
                System.out.print(ja[i][j] + " ");
            }
            System.out.println("\n");
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\JaggedArray.java

1 2 3



1 2 3
```

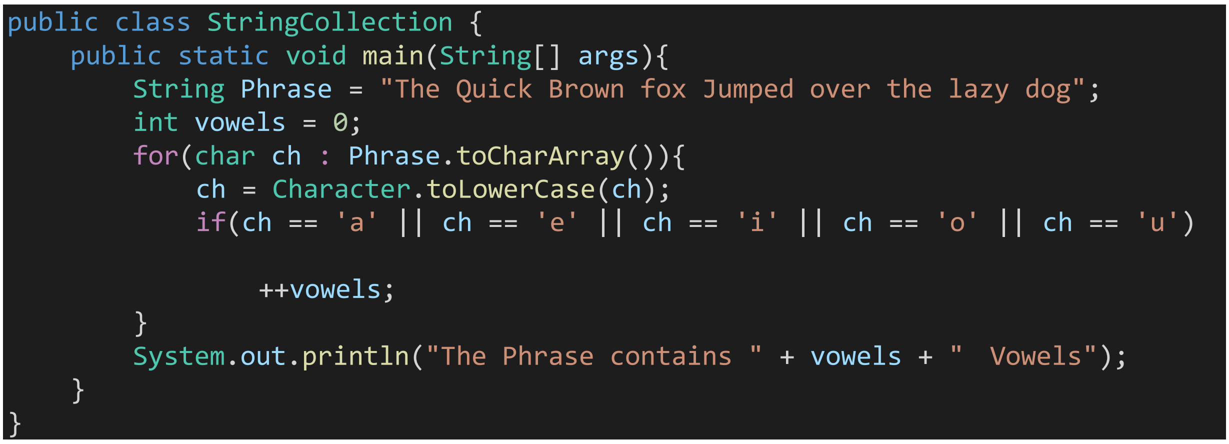
Q. Array 1 Example

```java
import java.util.Arrays;
public class ArrayExample{
    public static void main(String[] args) {
        int arr[]=new int [5];

        for(int i=5;i>0;i--)
            arr[5-i] = i;

        Arrays.sort(arr);

        for(int i=0;i<5;++i)
            System.out.print(arr[i]);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\ArrayExample.java

12345
```

Q. String Collection

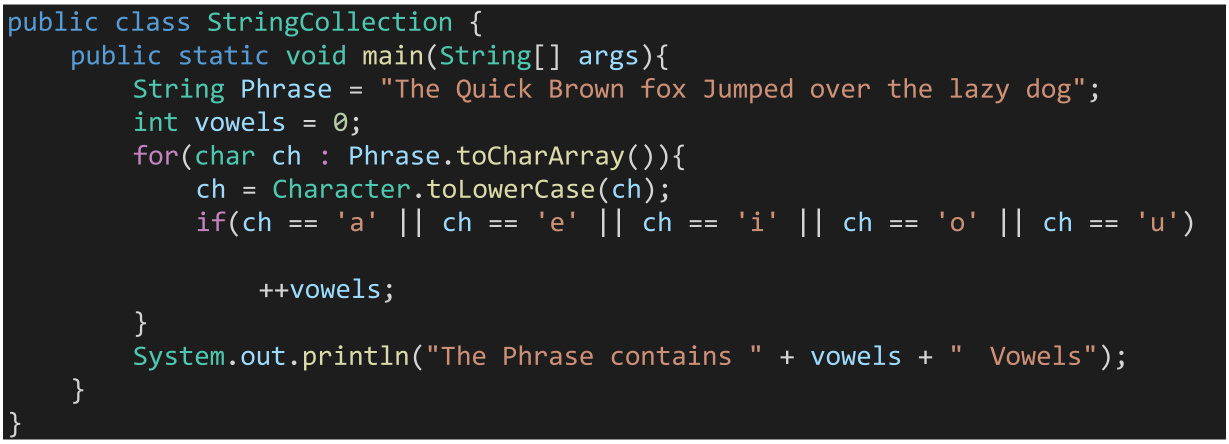


```java
public class StringCollection {
    public static void main(String[] args){
        String Phrase = "The Quick Brown fox Jumped over the lazy dog";
        int vowels = 0;
        for(char ch : Phrase.toCharArray()){
            ch = Character.toLowerCase(ch);
            if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')

                ++vowels;
        }
        System.out.println("The Phrase contains " + vowels + "  Vowels");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\StringCollection.java

The Phrase contains 12 Vowels
```

Q. Count Vowels



```java
public class StringCollection {
    public static void main(String[] args){
        String Phrase = "The Quick Brown fox Jumped over the lazy dog";
        int vowels = 0;
        for(char ch : Phrase.toCharArray()){
            ch = Character.toLowerCase(ch);
            if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')

                ++vowels;
        }
        System.out.println("The Phrase contains " + vowels + "  Vowels");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\StringCollection.java

The Phrase contains 12 Vowels
```

Q. Test String

```java
public class TestString {
    public static void main(String[] args){
        String s1 = "Test";
        String s2 = "Test";
        String s3 = new String("Test");
        final String s4 = s3.intern();
        System.out.println(s1 == s2);
        System.out.println(s2 == s3);
        System.out.println(s3 == s4);
        System.out.println(s1 == s3);
        System.out.println(s1 == s4);

        System.out.println(s1.equals(s2));
        System.out.println(s2.equals(s3));
        System.out.println(s3.equals(s4));
        System.out.println(s1.equals(s4));
        System.out.println(s1.equals(s3));
    }
}
```

Output:

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\TestString.java

true

false

false

false

true

true

true

true

true

true
```

Q. Lucky Star

```java
public class LuckyStar {
    public static void main(String[] args) {
        String[] stars = {
            "Robert Redford" , "Marilyn Monroe",
            "Boris Karloff" , "Lassie",
            "Hopalong Cassidy", "Trigger"
            };
        System.out.println("Your lucky star for today is " + stars[(int)(stars.length*Math.random())]);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\LuckyStar.java

Your lucky star for today is Boris Karloff
```

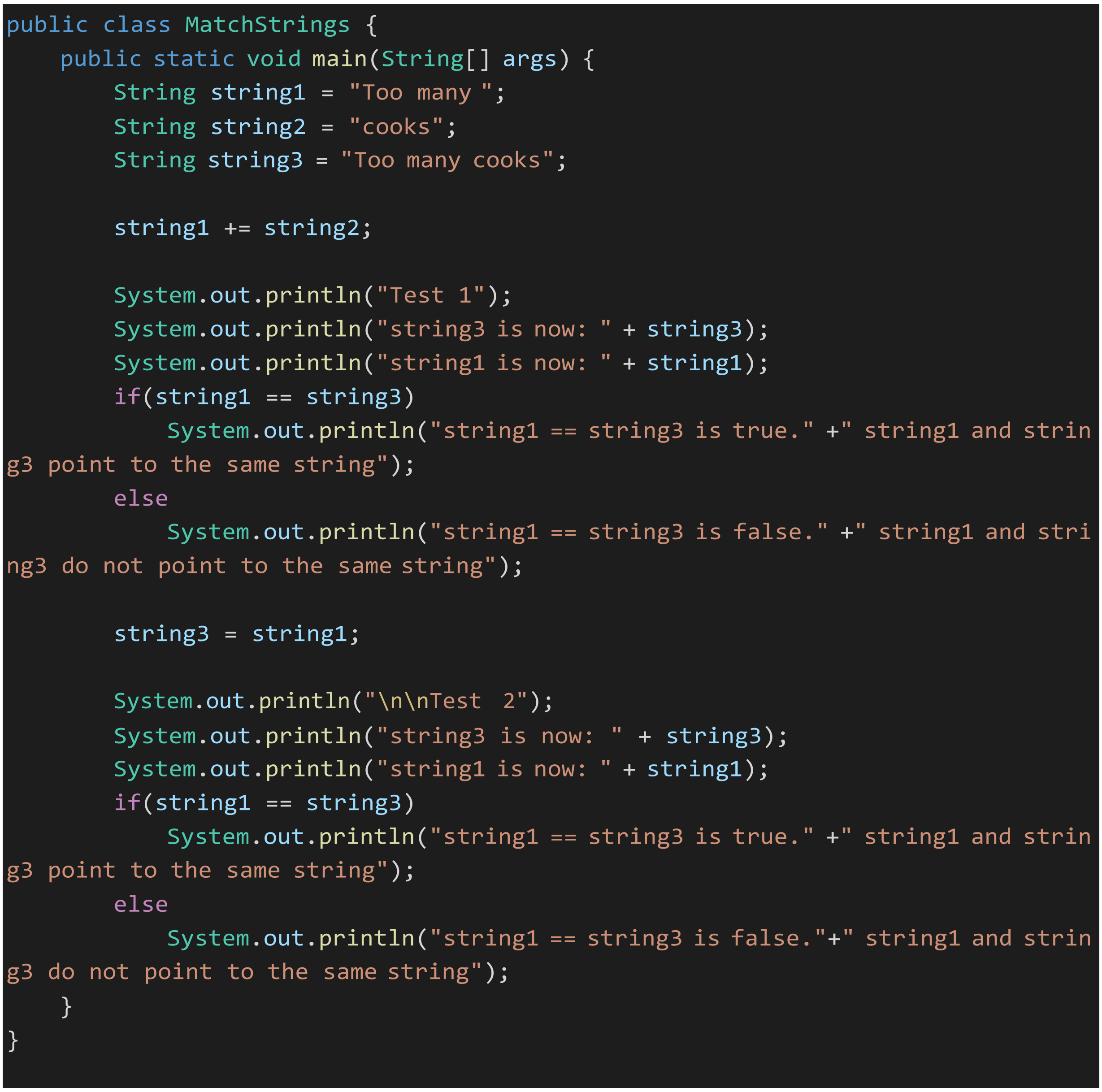
Q. Join String

```
public class JoinString {
    public static void main(String[] args) {
        String firstString = "Many ";
        String secondString = "hands ";
        String thirdString = "make light work";
        String myString;

        myString = firstString + secondString + thirdString;
        System.out.println(myString);

        int numHands = 99;
        myString = numHands + " " + secondString + thirdString;
        System.out.println(myString);


        myString = "fifty five is " + 5 + 5;
        System.out.println(myString);

        myString = 5 + 5 + " is ten ";
        System.out.println(myString);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\JoinString.java

Many hands make light work

99 hands make light work

fifty five is 55

10 is ten
```

Q. Match String



```java
public class MatchStrings {
    public static void main(String[] args) {
        String string1 = "Too many ";
        String string2 = "cooks";
        String string3 = "Too many cooks";

        string1 += string2;

        System.out.println("Test 1");
        System.out.println("string3 is now: " + string3);
        System.out.println("string1 is now: " + string1);
        if(string1 == string3)
            System.out.println("string1 == string3 is true." +" string1 and string3 point to the same string");
        else
            System.out.println("string1 == string3 is false." +" string1 and string3 do not point to the same string");

        string3 = string1;

        System.out.println("\n\nTest  2");
        System.out.println("string3 is now: " + string3);
        System.out.println("string1 is now: " + string1);
        if(string1 == string3)
            System.out.println("string1 == string3 is true." +" string1 and string3 point to the same string");
        else
            System.out.println("string1 == string3 is false."+" string1 and string3 do not point to the same string");
    }
}
```

Output :

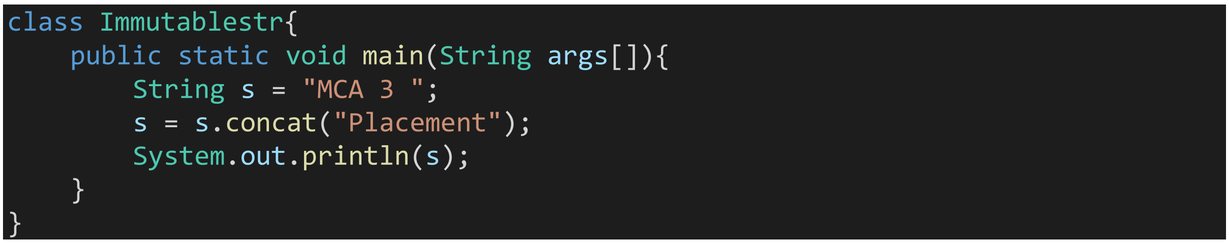
```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\MatchStrings.java

Test 1

string3 is now: Too many cooks
```

```
string1 is now: Too many cooks

string1 == string3 is false. string1 and string3 do not point to the same string
```

Test 2

```
string3 is now: Too many cooks

string1 is now: Too many cooks

string1 == string3 is true. string1 and string3 point to the same string
```

Q. Match String (immutable string)

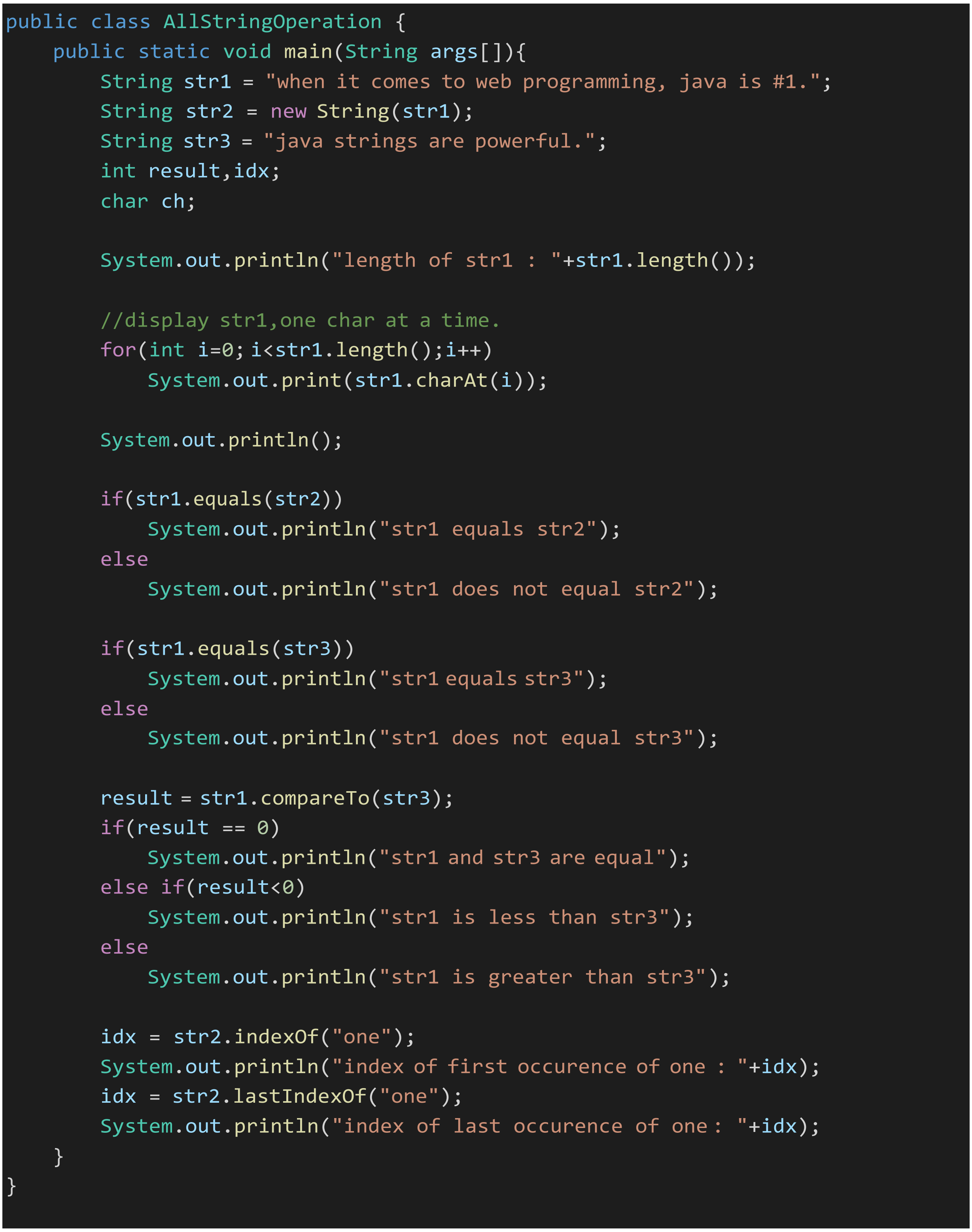


```
class Immutablestr{
    public static void main(String args[]){
        String s = "MCA 3 ";
        s = s.concat("Placement");
        System.out.println(s);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Immutablestr.java

MCA 3 Placement
```

Q. All Method of String



```
public class AllStringOperation {
    public static void main(String args[]){
        String str1 = "when it comes to web programming, java is #1.";
        String str2 = new String(str1);
        String str3 = "java strings are powerful.";
        int result,idx;
        char ch;

        System.out.println("length of str1 : "+str1.length());

        //display str1,one char at a time.
        for(int i=0; i<str1.length();i++)
            System.out.print(str1.charAt(i));

        System.out.println();

        if(str1.equals(str2))
            System.out.println("str1 equals str2");
        else
            System.out.println("str1 does not equal str2");

        if(str1.equals(str3))
            System.out.println("str1 equals str3");
        else
            System.out.println("str1 does not equal str3");

        result = str1.compareTo(str3);
        if(result == 0)
            System.out.println("str1 and str3 are equal");
        else if(result<0)
            System.out.println("str1 is less than str3");
        else
            System.out.println("str1 is greater than str3");

        idx = str2.indexOf("one");
        System.out.println("index of first occurence of one : "+idx);
        idx = str2.lastIndexOf("one");
        System.out.println("index of last occurence of one : "+idx);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\AllStringOperation.java

length of str1 : 45

when it comes to web programming, java is #1.

str1 equals str2

str1 does not equal str3

str1 is greater than str3

index of first occurence of one : -1

index of last occurence of one : -1
```

Q. Array of String

```java
public class ArrayString {
    public static void main(String args[]){
        String strs [] = { "this","is","a","test."};
        System.out.print("original array : ");
        for(String s : strs)
            System.out.print(s+"  ");

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

        strs[1] = "was";
        strs[3] = "test , tool";
        System.out.print("Modified array : ");
        for(String s : strs)
            System.out.print(s+"  ");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\ArrayString.java

original array : this is a test.



Modified array : this was a test , tool
```

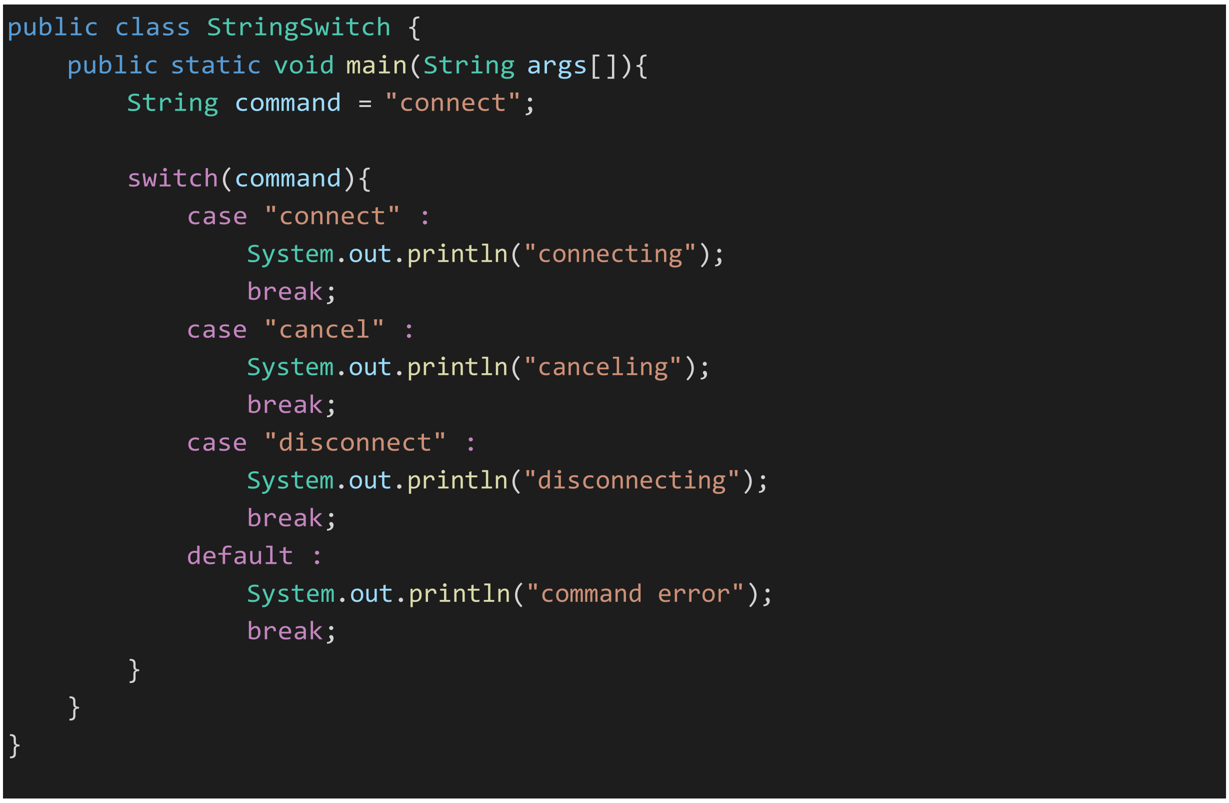
Q. Sub-String

```
public class SubString {
    public static void main(String args[]){
        String orgstr = "java makes the web move.";

        String substr = orgstr.substring(5,18);

        System.out.println("orgstr : "+orgstr);
        System.out.println("substr : "+substr);
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\SubString.java

orgstr : java makes the web move.

substr : makes the web
```

Q. String Switch

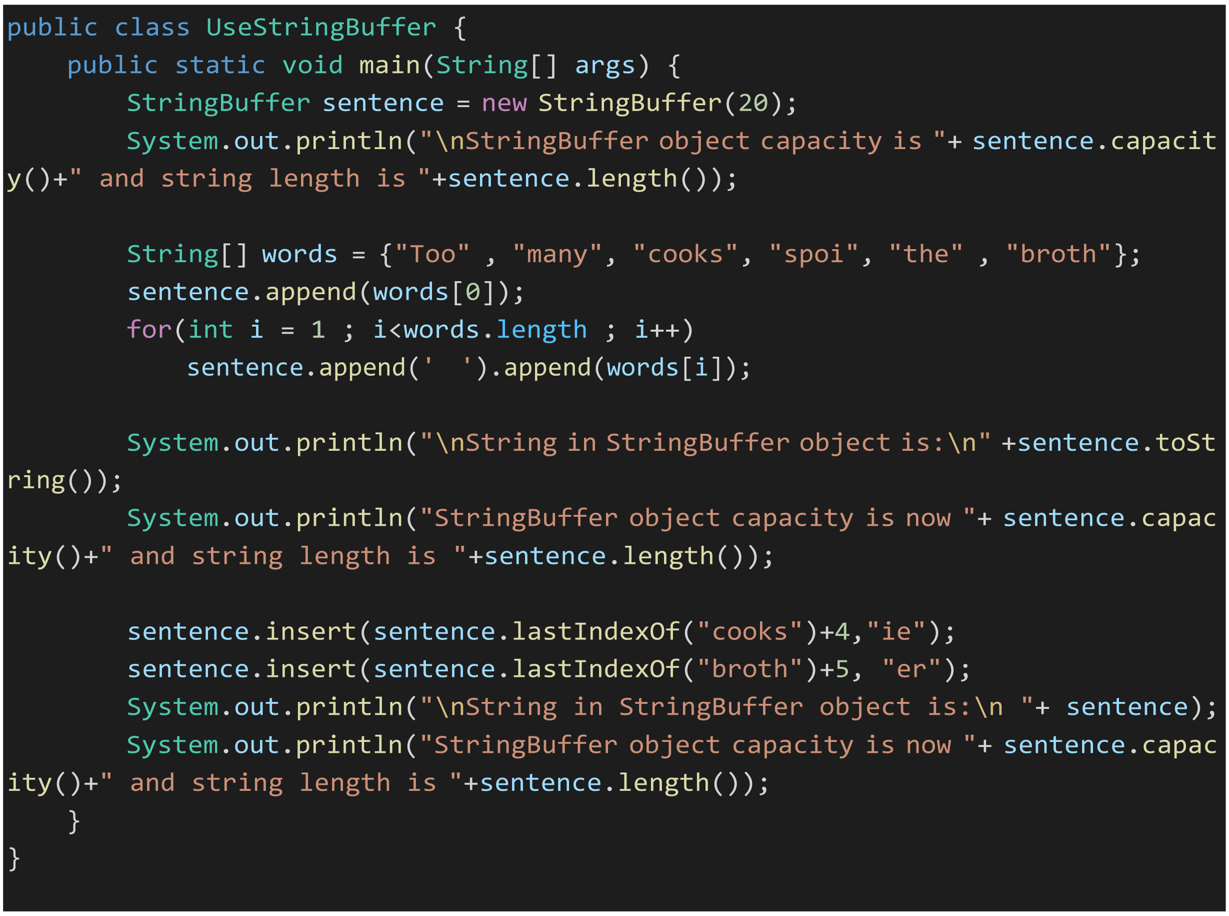


```java
public class StringSwitch {
    public static void main(String args[]){
        String command = "connect";

        switch(command){
            case "connect" :
                System.out.println("connecting");
                break;
            case "cancel" :
                System.out.println("canceling");
                break;
            case "disconnect" :
                System.out.println("disconnecting");
                break;
            default :
                System.out.println("command error");
                break;
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\StringSwitch.java

Connecting
```

Q. Using a StringBuffer Object to Assemble a String



```java
public class UseStringBuffer {
    public static void main(String[] args) {
        StringBuffer sentence = new StringBuffer(20);
        System.out.println("\nStringBuffer object capacity is "+ sentence.capacity()+" and string length is "+sentence.length());

        String[] words = {"Too" , "many", "cooks", "spoi", "the" , "broth"};
        sentence.append(words[0]);
        for(int i = 1 ; i<words.length ; i++)
            sentence.append(' ').append(words[i]);

        System.out.println("\nString in StringBuffer object is:\n" +sentence.toString());
        System.out.println("StringBuffer object capacity is now "+ sentence.capacity()+" and string length is "+sentence.length());

        sentence.insert(sentence.lastIndexOf("cooks")+4,"ie");
        sentence.insert(sentence.lastIndexOf("broth")+5, "er");
        System.out.println("\nString in StringBuffer object is:\n "+ sentence);
        System.out.println("StringBuffer object capacity is now "+ sentence.capacity()+" and string length is "+sentence.length());
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\UseStringBuffer.java

StringBuffer object capacity is 20 and string length is 0

String in StringBuffer object is:
```

```
Too many cooks spoi the broth

StringBuffer object capacity is now 42 and string length is 29


String in StringBuffer object is:

 Too many cookies spoi the brother

StringBuffer object capacity is now 42 and string length is 33
```

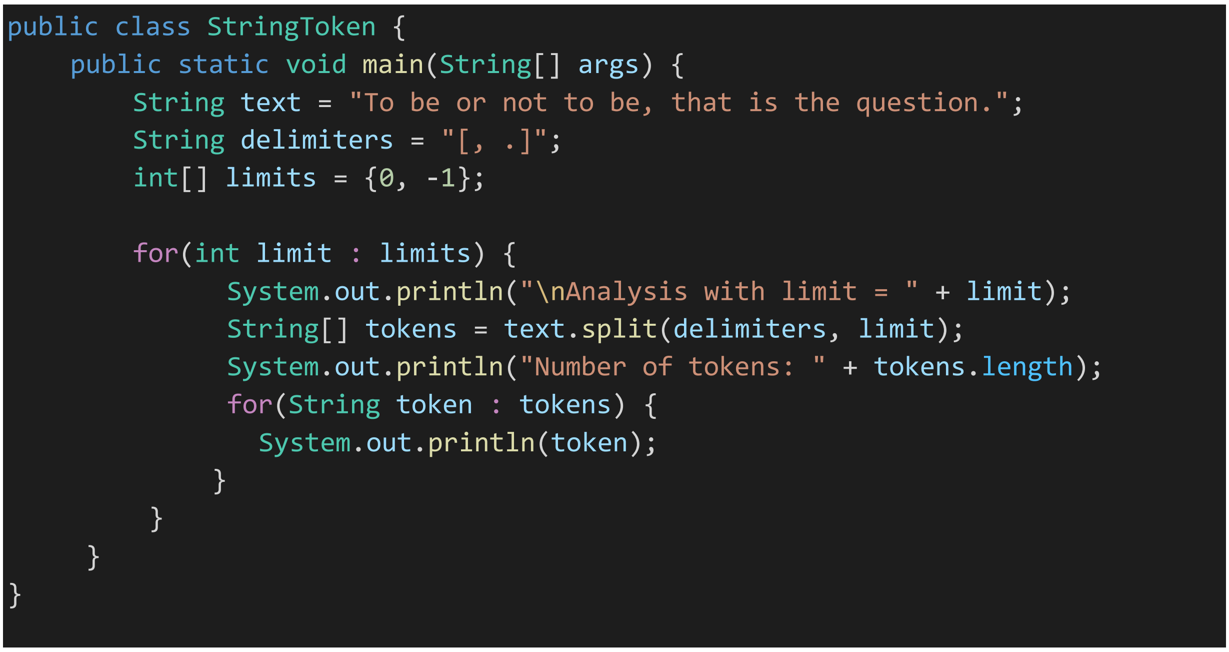
Q. Exciting Concordance Entries

```
public class FindCharacter {
    public static void main(String[] args) {

        String text = "To be or not to be, that is the question;" + " Whether ‘ti
s nobler in the mind to suffer" + " the slings and arrows of outrageous fortune,"
         + " or to take arms against a sea of troubles," + " and by opposing end
them?";
        int andCount = 0;
        int theCount = 0;
        int index = -1;
        String andStr = "and";
        String theStr = "the";

        index =text.indexOf(andStr);
        while(index >= 0) {
            ++andCount;
            index += andStr.length();
            index = text.indexOf(andStr, index);
        }

        index = text.lastIndexOf(theStr);
        while(index >= 0) {
            ++theCount;
            index -= theStr.length();
            index = text.lastIndexOf(theStr, index);
        }
       System.out.println("The text contains " + andCount + " ands\n" + "The text
 contains " + theCount + " thes");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\FindCharacter.java

The text contains 2 ands

The text contains 5 thes
```

Q. String Tokenizing



```
public class StringToken {
    public static void main(String[] args) {
        String text = "To be or not to be, that is the question.";
        String delimiters = "[, .]";
        int[] limits = {0, -1};

        for(int limit : limits) {
              System.out.println("\nAnalysis with limit = " + limit);
              String[] tokens = text.split(delimiters, limit);
              System.out.println("Number of tokens: " + tokens.length);
              for(String token : tokens) {
                System.out.println(token);
             }
         }
     }
}
```

Output :

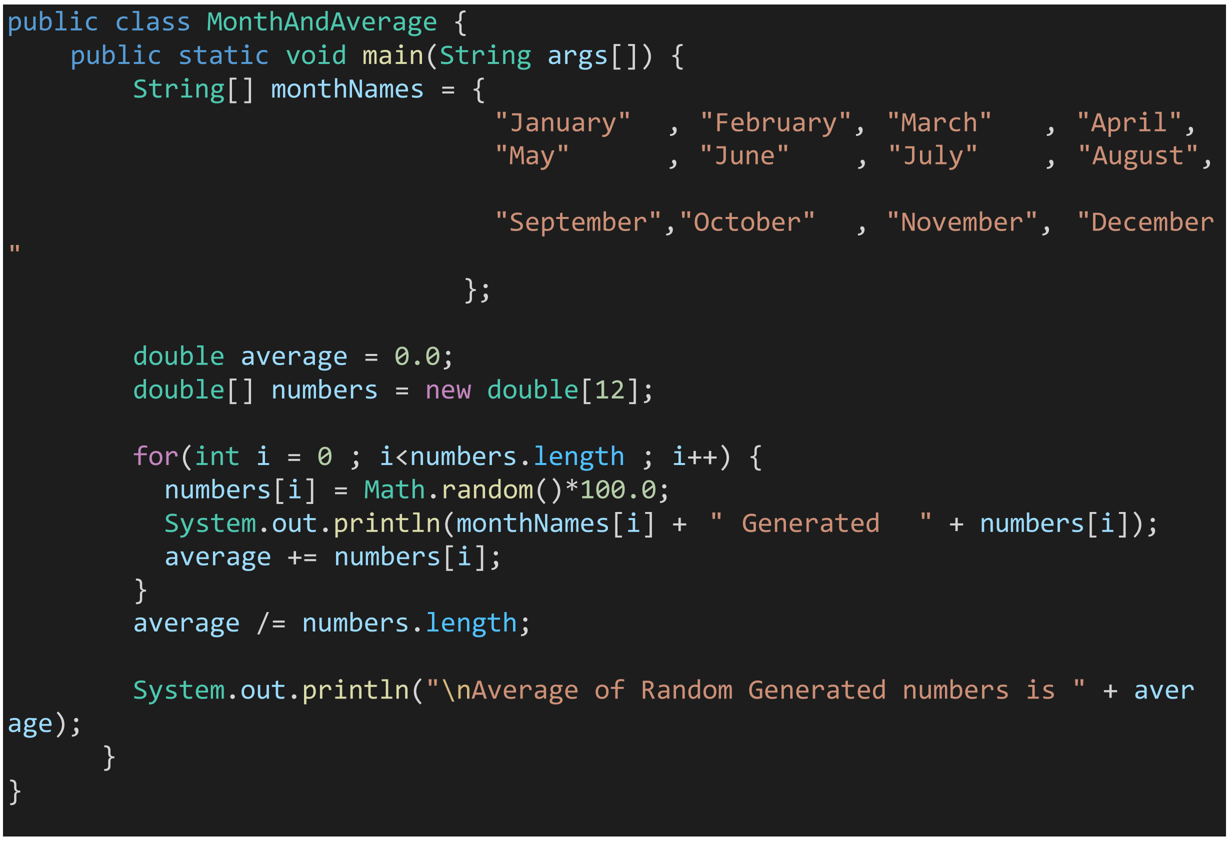
```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\StringToken.java


Analysis with limit = 0

Number of tokens: 11

To

be

or

not

to

be
```

```
that
is
the
question

Analysis with limit = -1
Number of tokens: 12
To
be
or
not
to
be

that
is
the
question
```

Excercise - 2

Q. Exercise

- **Month And Average**



```
public class MonthAndAverage {
    public static void main(String args[]) {
        String[] monthNames = {
                                "January"  , "February", "March"    , "April",
                                "May"      , "June"    , "July"     , "August",

                                "September","October"  , "November", "December
"
                              };

        double average = 0.0;
        double[] numbers = new double[12];

        for(int i = 0 ; i<numbers.length ; i++) {
          numbers[i] = Math.random()*100.0;
          System.out.println(monthNames[i] + " Generated  " + numbers[i]);
          average += numbers[i];
        }
        average /= numbers.length;

        System.out.println("\nAverage of Random Generated numbers is " + aver
age);
      }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\23> java .\MonthAndAverage.java

January Generated  14.796351800091923

February Generated 39.711269832223785

March Generated 58.32386495973172

April Generated  19.083891467683557

May Generated   78.52264624324705

June Generated   72.46307769949362
```

July Generated 64.2760130349091

August Generated 51.97756817789709

September Generated 17.767136593640963

October Generated 71.2641366807655

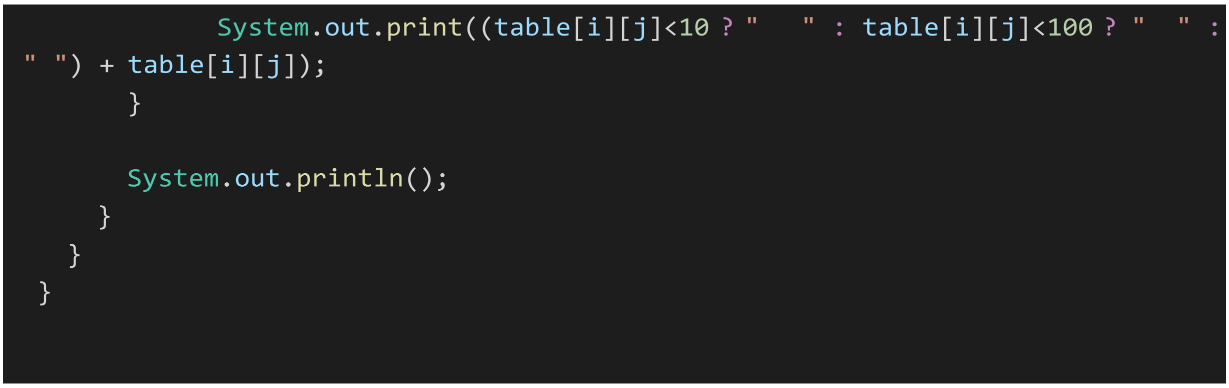
November Generated   82.89356693449919

December Generated   49.55683721304006

Average of Random Generated numbers is 51.71969671976862

- **Tables**

```
public class Tables {
    final static int TABLESIZE = 12;
    public static void main(String[]args) {
      int[][] table = new int[TABLESIZE][TABLESIZE];

      for(int i = 0 ; i<table.length ; i++) {
        for(int j = 0 ; j<table[i].length ; j++)
            table[i][j] = (i+1)*(j+1);
      }

      System.out.print("        ");

      for(int j = 1 ; j<=table[0].length ; j++) {
        System.out.print((j<10 ? "   ": "  ") + j);
      }

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

      for(int i=0; i<table.length; i++) {
        System.out.print("Row" + (i<9 ? "  ":" ") + (i+1) + ":");

        for(int j=0; j<table[i].length; j++) {
```



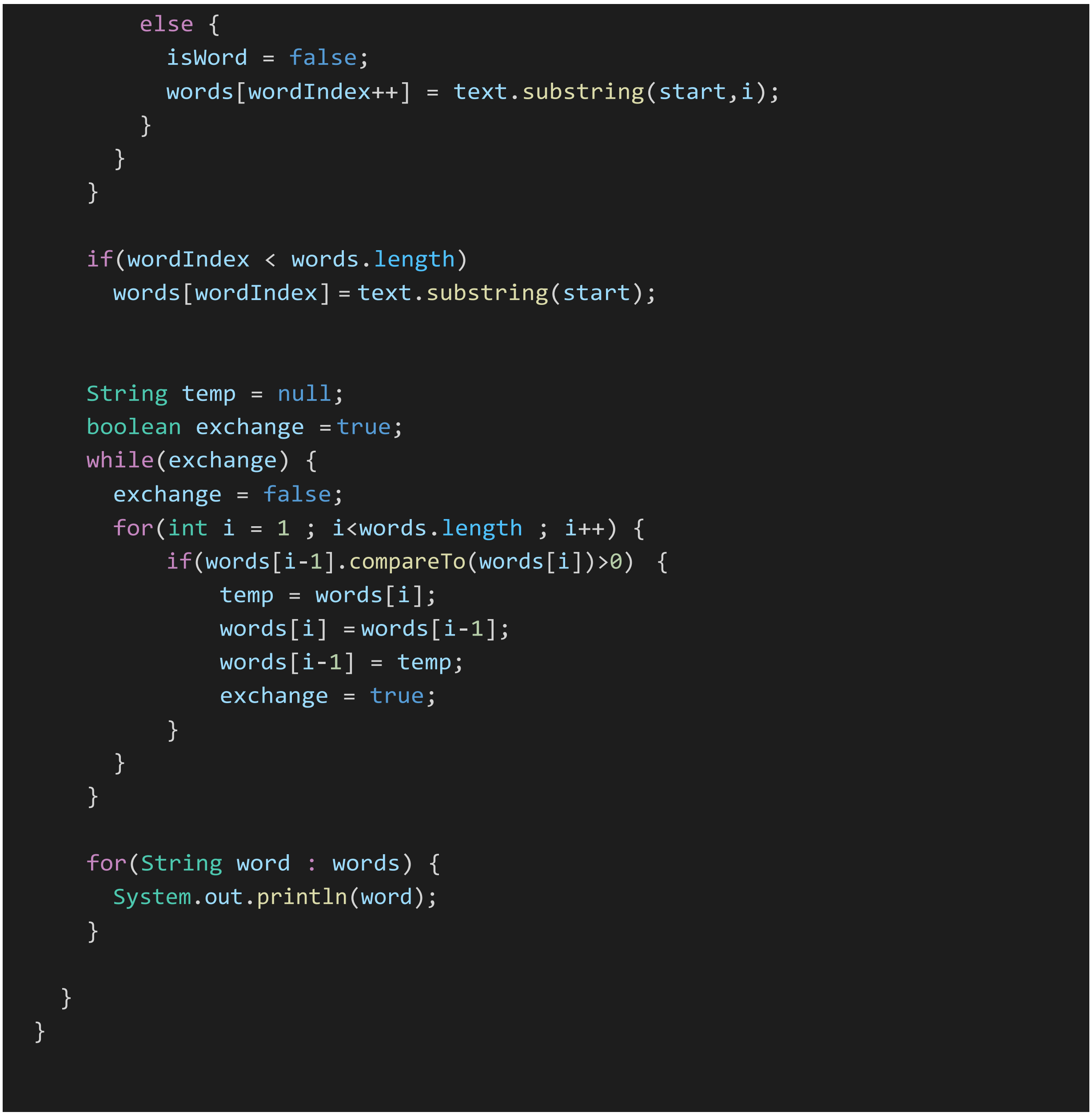
```
              System.out.print((table[i][j]<10 ? "   " : table[i][j]<100 ? "  " :
" ") + table[i][j]);
        }

        System.out.println();
      }
    }
  }
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\23> java .\Tables.java

      1 2 3 4 5 6 7 8 9 10 11 12

------------------------------------------------------------------

Row  1:   1   2   3   4   5 6 7 8 9 10 11 12

Row  2:   2   4   6   8  10 12 14 16 18 20 22 24

Row  3:   3   6   9  12  15 18 21 24 27 30 33 36

Row  4:   4   8 12   16 20 24 28 32 36 40 44 48

Row  5:   5  10  15  20  25  30  35  40  45  50  55  60

Row  6:   6  12  18  24  30  36  42  48  54  60  66  72

Row  7:   7  14  21  28  35  42  49  56  63  70  77  84

Row  8:   8  16  24  32  40  48  56  64  72  80  88  96

Row  9:   9  18  27  36  45  54  63  72  81  90  99 108

Row 10:  10  20  30  40  50  60  70  80 90 100 110 120

Row 11:  11  22  33  44  55  66  77  88 99 110 121 132

Row 12:  12  24  36  48  60  72  84  96 108 120 132 144
```

- **Sort**

```java
public class Sort {
    public static void main(String args[]) {

      String text = "Lorem Ipsum is simply dummy text of the printing and typesetting industry " +
                    "It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout ";

      int count = 0;
      boolean isWord = false;
      for (int i = 0 ; i<text.length() ; i++) {
        if(isWord) {
          if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\'')
            continue;
          else
            isWord = false;
        }
        else if(Character.isLetter(text.charAt(i))) {

            count++;
            isWord = true;
        }
      }


      String[] words = new String[count];


      int start = 0;
      int wordIndex = 0;

      isWord = false;
      for (int i = 0 ; i<text.length() ; i++) {
        if(!isWord) {
          if(Character.isLetter(text.charAt(i))) {
            start = i;
            isWord = true;
          }
        }
        else {
          if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\'')
            continue;
```



```
    else {
      isWord = false;
      words[wordIndex++] = text.substring(start,i);
    }
   }
  }

  if(wordIndex < words.length)
   words[wordIndex] = text.substring(start);


  String temp = null;
  boolean exchange =true;
  while(exchange) {
   exchange = false;
   for(int i = 1 ; i<words.length ; i++) {
     if(words[i-1].compareTo(words[i])>0)  {
       temp = words[i];
       words[i] =words[i-1];
       words[i-1] = temp;
       exchange = true;
     }
   }
  }

  for(String word : words) {
   System.out.println(word);
  }

 }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\23> java .\Sort.java

Ipsum

It

Lorem
```

a

a

a

and

at

be

by

content

distracted

dummy

established

fact

industry

is

is

its

layout

long

looking

of

of

page

printing

readable

reader

simply

text

that

the

the

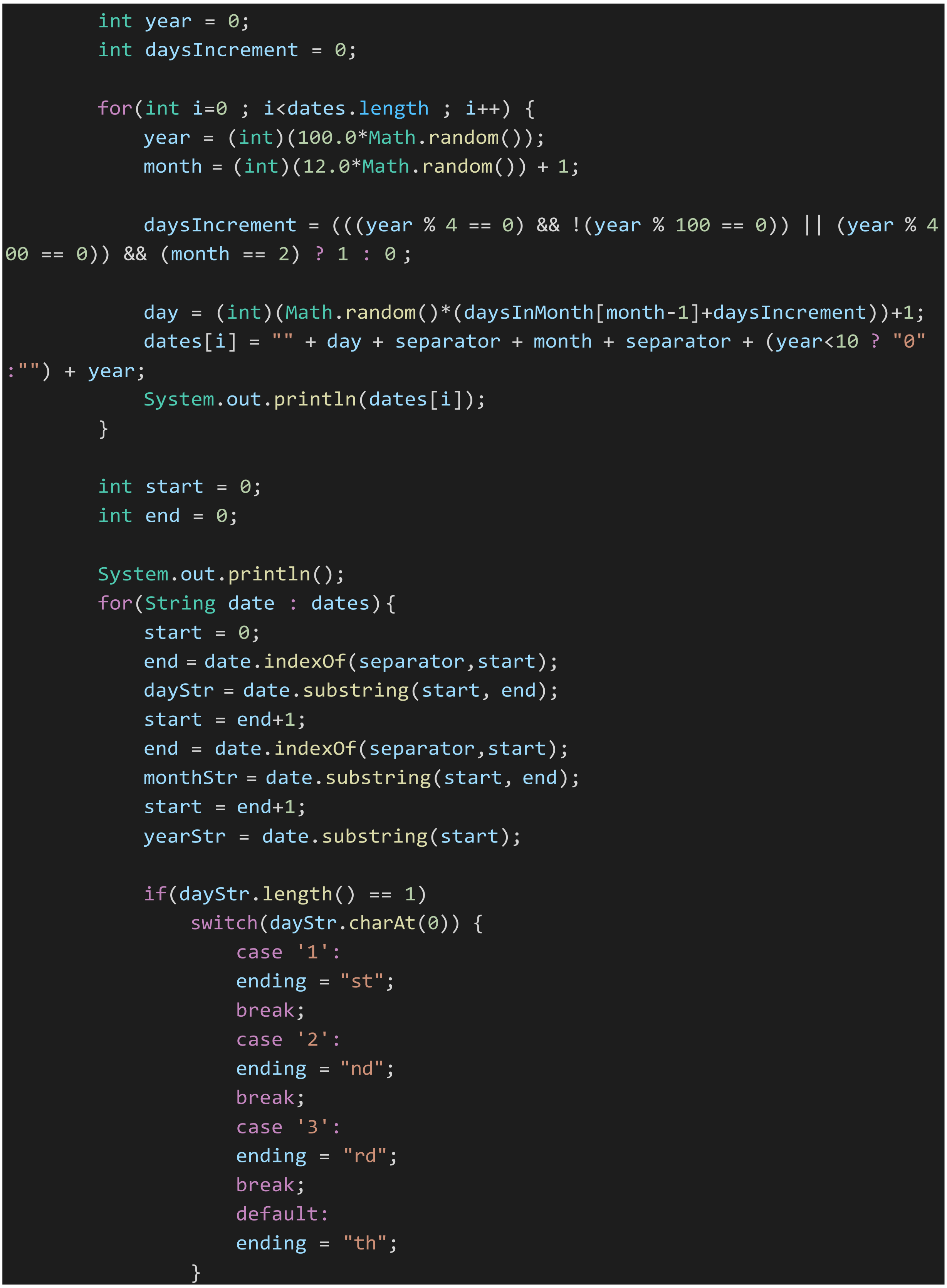
typesetting

when

will

- **Dates**

```
public class Date {
    public static void main(String args[]) {
        String[] dates = new String[10];
        String[] monthNames = {
                "January", "February", "March", "April",
                "May", "June", "July", "August", "September",
                "October", "November", "December"
                };

        int[] daysInMonth = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        String ending = "th";
        String dayStr = null;
        String monthStr =null;
        String yearStr = null;
        char separator = '/';
        int day = 0;
        int month = 0;
```



```
        int year = 0;
        int daysIncrement = 0;

        for(int i=0 ; i<dates.length ; i++) {
            year = (int)(100.0*Math.random());
            month = (int)(12.0*Math.random()) + 1;

            daysIncrement = (((year % 4 == 0) && !(year % 100 == 0)) || (year % 4
00 == 0)) && (month == 2) ? 1 : 0 ;

            day = (int)(Math.random()*(daysInMonth[month-1]+daysIncrement))+1;
            dates[i] = "" + day + separator + month + separator + (year<10 ? "0"
:"") + year;
            System.out.println(dates[i]);
        }

        int start = 0;
        int end = 0;

        System.out.println();
        for(String date : dates){
            start = 0;
            end = date.indexOf(separator,start);
            dayStr = date.substring(start, end);
            start = end+1;
            end = date.indexOf(separator,start);
            monthStr = date.substring(start, end);
            start = end+1;
            yearStr = date.substring(start);

            if(dayStr.length() == 1)
                switch(dayStr.charAt(0)) {
                    case '1':
                    ending = "st";
                    break;
                    case '2':
                    ending = "nd";
                    break;
                    case '3':
                    ending = "rd";
                    break;
                    default:
                    ending = "th";
                }
```

```java
            else if(dayStr.charAt(0) == '1')
                ending = "th";
            else
                switch(dayStr.charAt(1)) {
                    case '1':
                    ending = "st";
                    break;
                    case '2':
                    ending = "nd";
                    break;
                    case '3':
                    ending = "rd";
                    break;
                    default:
                    ending = "th";
                }


            System.out.println(dayStr + ending + " " +
                monthNames[monthStr.length() == 1 ? monthStr.charAt(0) -
 '1': 9 + monthStr.charAt(1) - '0'] + " " + "19" + yearStr);
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\23> java .\Date.java

14/11/55

9/2/85

6/1/93

28/12/07

21/2/78

12/2/04

3/11/13
```

31/1/31

1/6/85

6/1/61

14th November 1955

9th February 1985

6th January 1993

28th December 1907

21st February 1978

12th February 1904

3rd November 1913

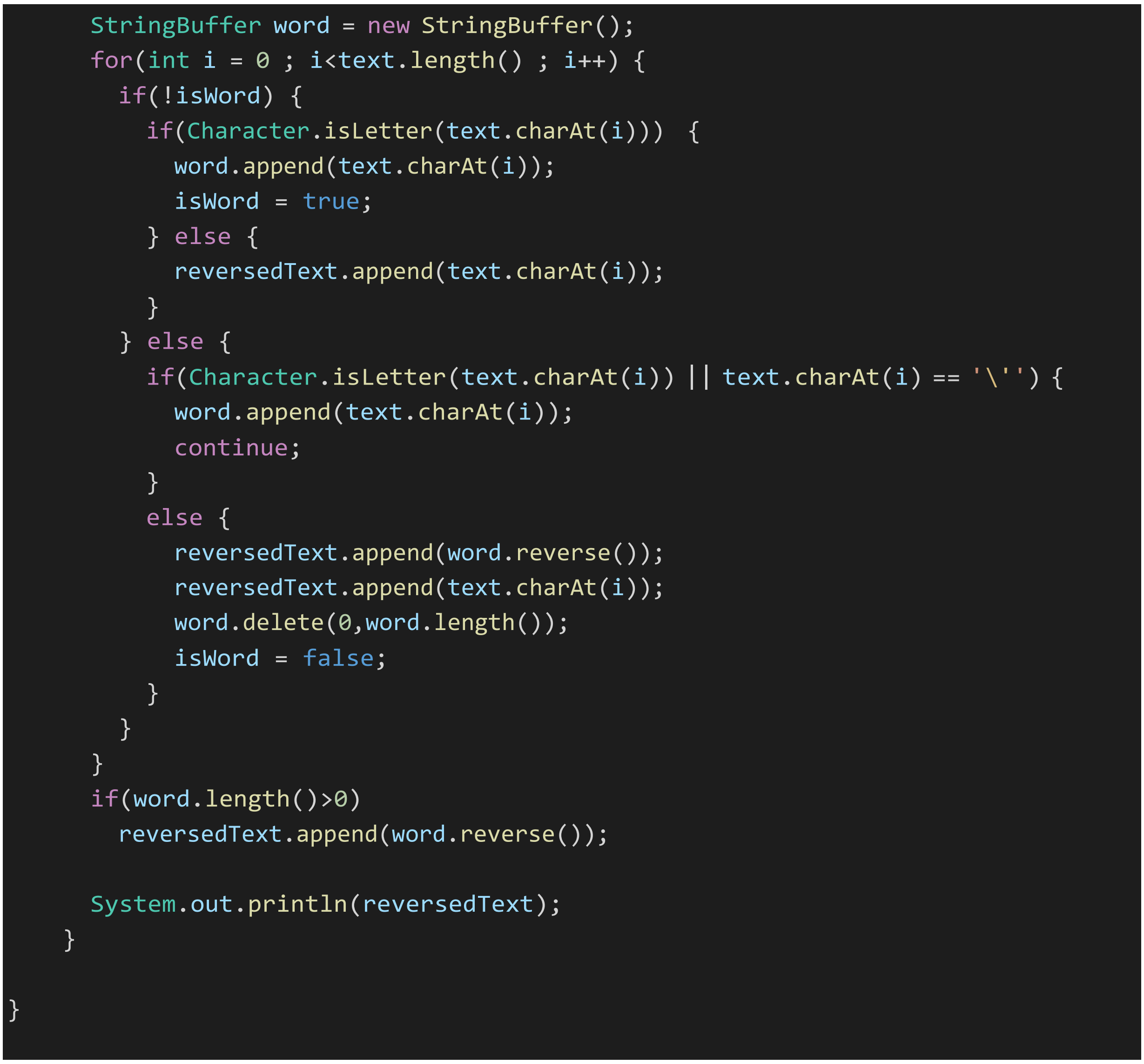
31st January 1931

1st June 1985

6th January 1961

- **Reverse**

```
public class Reverse {

    public static void main(String args[]) {
      String text = "Lorem Ipsum is simply dummy text of the printing and typesetting industry " +
      "\nIt is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout ";

      boolean isWord = false;
      int start = 0;
      StringBuffer reversedText = new StringBuffer();
```



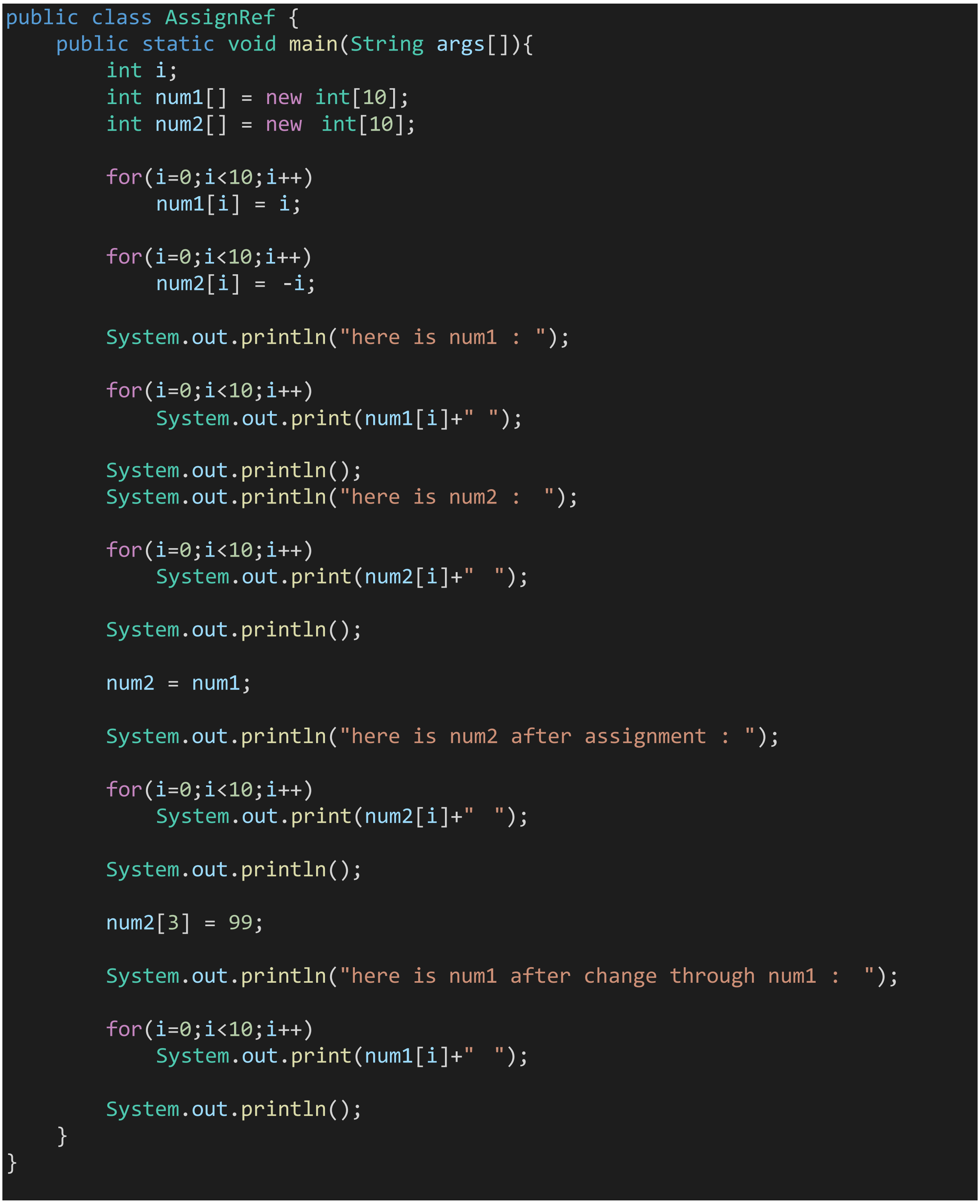
```
      StringBuffer word = new StringBuffer();
      for(int i = 0 ; i<text.length() ; i++) {
        if(!isWord) {
          if(Character.isLetter(text.charAt(i)))  {
            word.append(text.charAt(i));
            isWord = true;
          } else {
            reversedText.append(text.charAt(i));
          }
        } else {
          if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\'') {
            word.append(text.charAt(i));
            continue;
          }
          else {
            reversedText.append(word.reverse());
            reversedText.append(text.charAt(i));
            word.delete(0,word.length());
            isWord = false;
          }
        }
      }
      if(word.length()>0)
        reversedText.append(word.reverse());

      System.out.println(reversedText);
    }

}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\23> java .\Reverse.java

meroL muspI si ylpmis ymmud txet fo eht gnitnirp dna gnittesepyt yrtsudni

tI si a gnol dehsilbatse tcaf taht a redaer lliw eb detcartsid yb eht elbadaer
tnetnoc fo a egap nehw gnikool ta sti tuoyal
```

Class Work - 3

Q. Reference Assignment

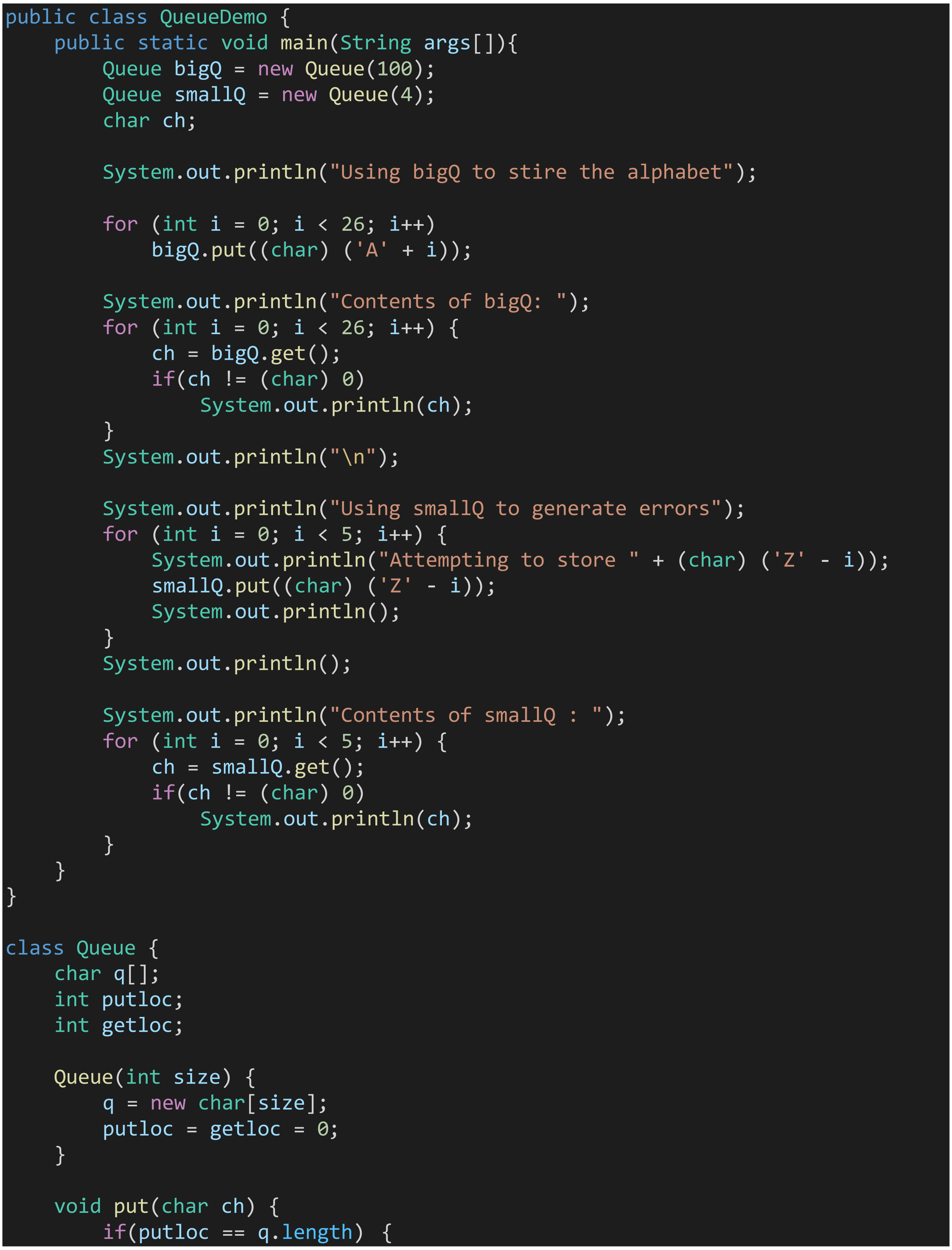


```
public class AssignRef {
    public static void main(String args[]){
        int i;
        int num1[] = new int[10];
        int num2[] = new  int[10];

        for(i=0;i<10;i++)
            num1[i] = i;

        for(i=0;i<10;i++)
            num2[i] = -i;

        System.out.println("here is num1 : ");

        for(i=0;i<10;i++)
            System.out.print(num1[i]+" ");

        System.out.println();
        System.out.println("here is num2 :  ");

        for(i=0;i<10;i++)
            System.out.print(num2[i]+"  ");

        System.out.println();

        num2 = num1;

        System.out.println("here is num2 after assignment : ");

        for(i=0;i<10;i++)
            System.out.print(num2[i]+"  ");

        System.out.println();

        num2[3] = 99;

        System.out.println("here is num1 after change through num1 :  ");

        for(i=0;i<10;i++)
            System.out.print(num1[i]+"  ");

        System.out.println();
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\AssignRef.java

here is num1 :

0 1 2 3 4 5 6 7 8 9

here is num2 :

0 -1 -2 -3 -4 -5 -6 -7 -8 -9

here is num2 after assignment :

0 1 2 3 4 5 6 7 8 9

here is num1 after change through num1 :

0 1 2 99 4 5 6 7 8 9
```

Q. Queue Demo



```java
public class QueueDemo {
    public static void main(String args[]){
        Queue bigQ = new Queue(100);
        Queue smallQ = new Queue(4);
        char ch;

        System.out.println("Using bigQ to stire the alphabet");

        for (int i = 0; i < 26; i++)
            bigQ.put((char) ('A' + i));

        System.out.println("Contents of bigQ: ");
        for (int i = 0; i < 26; i++) {
            ch = bigQ.get();
            if(ch != (char) 0)
                System.out.println(ch);
        }
        System.out.println("\n");

        System.out.println("Using smallQ to generate errors");
        for (int i = 0; i < 5; i++) {
            System.out.println("Attempting to store " + (char) ('Z' - i));
            smallQ.put((char) ('Z' - i));
            System.out.println();
        }
        System.out.println();

        System.out.println("Contents of smallQ : ");
        for (int i = 0; i < 5; i++) {
            ch = smallQ.get();
            if(ch != (char) 0)
                System.out.println(ch);
        }
    }
}

class Queue {
    char q[];
    int putloc;
    int getloc;

    Queue(int size) {
        q = new char[size];
        putloc = getloc = 0;
    }

    void put(char ch) {
        if(putloc == q.length)  {
            System.out.println(" - Queue is Full.");
            return;
        }

        q[putloc++] = ch;
    }

    char get() {
        if(getloc == putloc) {
            System.out.println(" - Queue is Empty.");
            return (char) 0;
        }

        return q[getloc++];
    }
}
```

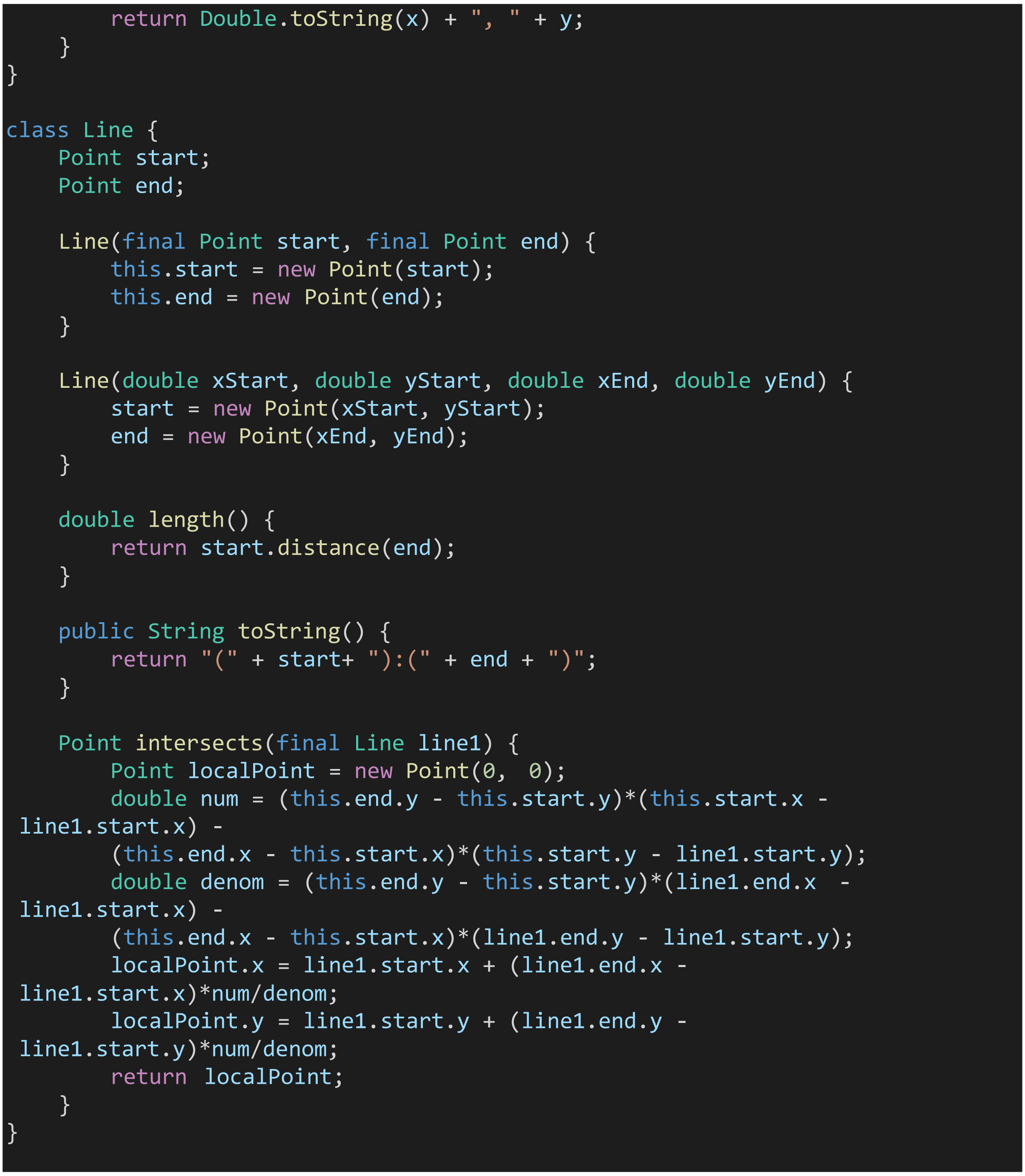
Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\QueueDemo.java

Using bigQ to stire the alphabet

Contents of bigQ:

A

B

C

D

E

F

G

H

I
```

```
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z


Using smallQ to generate errors
Attempting to store Z
Attempting to store Y


Attempting to store X


Attempting to store W


Attempting to store V

 - Queue is Full.



Contents of smallQ :

Z

Y

X

W

 - Queue is Empty.
```

Q. TryGeometry

```
import static java.lang.Math.sqrt;

public class TryGeometry {
    public static void main(String[] args) {
        Point start = new Point(0.0, 1.0);
        Point end = new Point(5.0, 6.0);
        System.out.println("Points created are " + start + " and " +  end);

        Line line1 = new Line(start, end);
        Line line2 = new Line(0.0, 3.0, 3.0, 0.0);
        System.out.println("Lines created are " + line1 + " and " + line2);

        System.out.println("Intersection is " + line2.intersects(line1));

        end.move(1.0, -5.0);
        System.out.println("Intersection is " +  line1.intersects(line2));

    }
}

class Point {
    // Coordinates of the point
    double x;
    double y;

    Point(double xVal, double yVal) {
        x = xVal;
        y = yVal;
    }

    Point(final Point oldPoint) {
        x = oldPoint.x;
        y =  oldPoint.y;
    }

    void move(double xDelta, double yDelta) {

        x += xDelta;
        y += yDelta;
    }

    double distance(final Point aPoint) {
        return sqrt((x - aPoint.x)*(x - aPoint.x) + (y - aPoint.y)*(y -
 aPoint.y));
    }


    public String toString() {
```

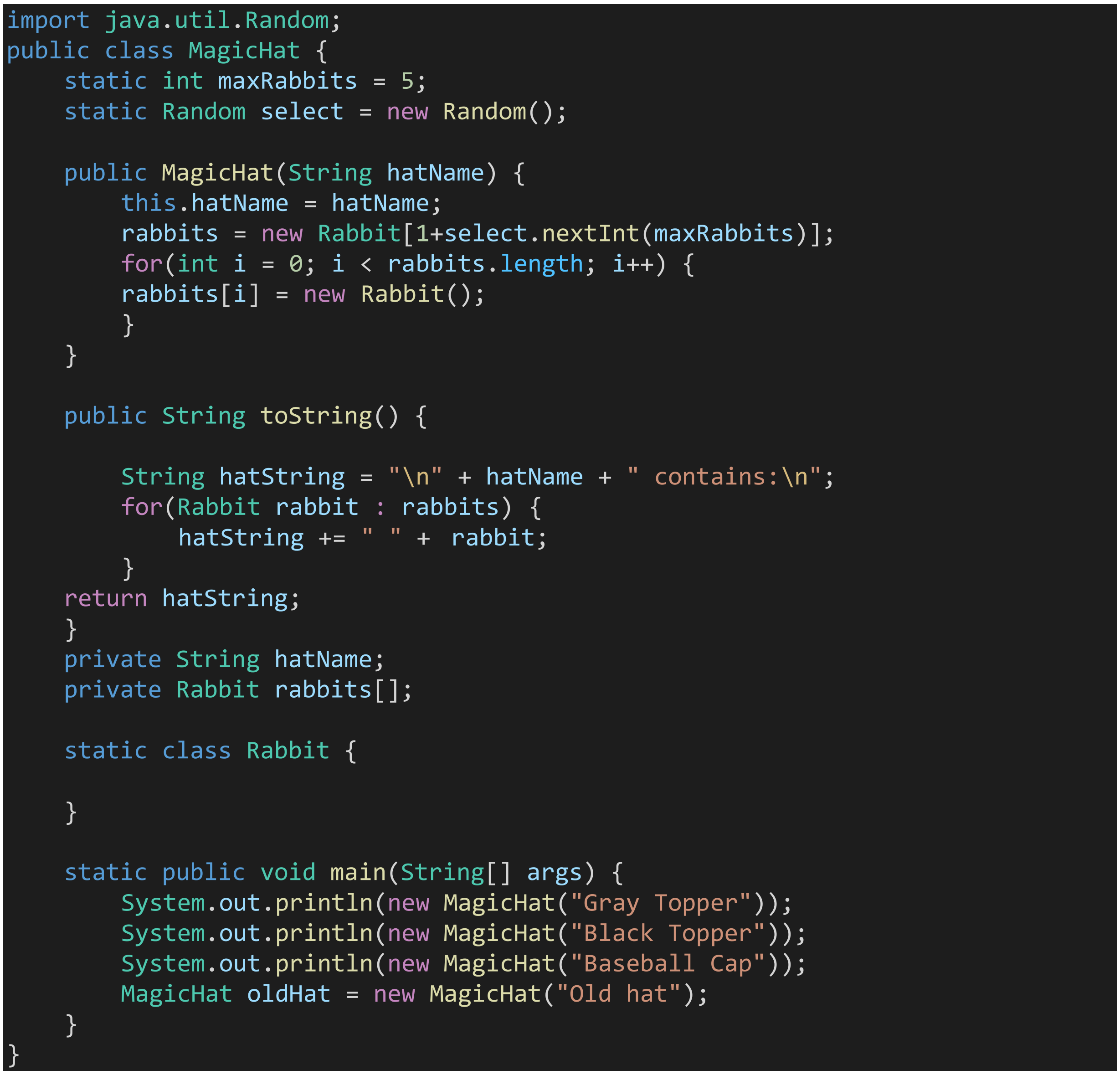


```
        return Double.toString(x) + ", " + y;
    }
}

class Line {
    Point start;
    Point end;

    Line(final Point start, final Point end) {
        this.start = new Point(start);
        this.end = new Point(end);
    }

    Line(double xStart, double yStart, double xEnd, double yEnd) {
        start = new Point(xStart, yStart);
        end = new Point(xEnd, yEnd);
    }

    double length() {
        return start.distance(end);
    }

    public String toString() {
        return "(" + start+ "):(" + end + ")";
    }

    Point intersects(final Line line1) {
        Point localPoint = new Point(0,  0);
        double num = (this.end.y - this.start.y)*(this.start.x -
 line1.start.x) -
        (this.end.x - this.start.x)*(this.start.y - line1.start.y);
        double denom = (this.end.y - this.start.y)*(line1.end.x  -
 line1.start.x) -
        (this.end.x - this.start.x)*(line1.end.y - line1.start.y);
        localPoint.x = line1.start.x + (line1.end.x -
 line1.start.x)*num/denom;
        localPoint.y = line1.start.y + (line1.end.y -
 line1.start.y)*num/denom;
        return localPoint;
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\trygeometry> java .\TryGeometry.java

Points created are 0.0, 1.0 and 5.0, 6.0

Lines created are (0.0, 1.0):(5.0, 6.0) and (0.0, 3.0):(3.0, 0.0)

Intersection is 1.0, 2.0

Intersection is 1.0, 2.0
```

Q. Rabbit Out of Hats



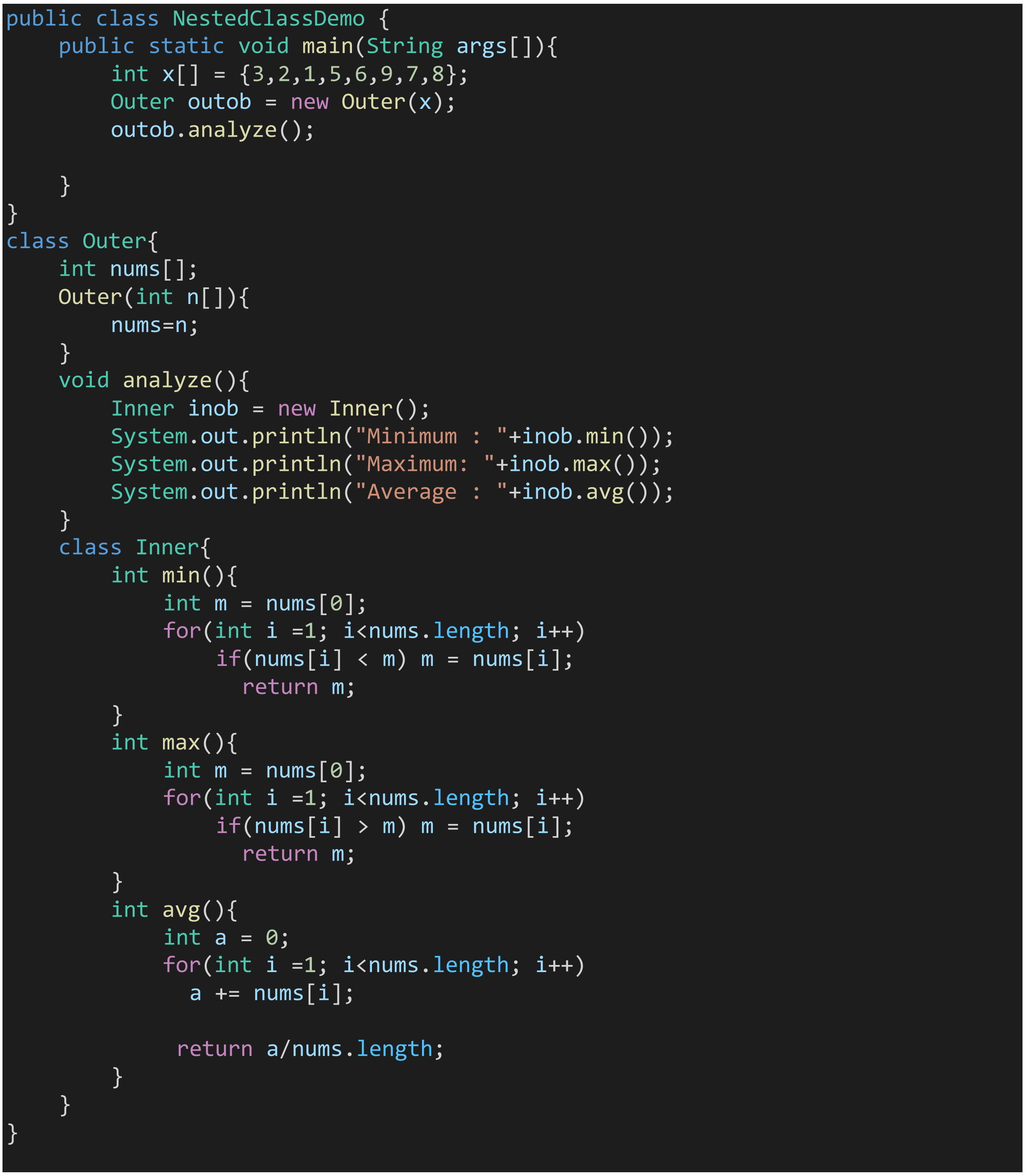
```java
import java.util.Random;
public class MagicHat {
    static int maxRabbits = 5;
    static Random select = new Random();

    public MagicHat(String hatName) {
        this.hatName = hatName;
        rabbits = new Rabbit[1+select.nextInt(maxRabbits)];
        for(int i = 0; i < rabbits.length; i++) {
        rabbits[i] = new Rabbit();
        }
    }

    public String toString() {

        String hatString = "\n" + hatName + " contains:\n";
        for(Rabbit rabbit : rabbits) {
            hatString += " " + rabbit;
        }
    return hatString;
    }
    private String hatName;
    private Rabbit rabbits[];

    static class Rabbit {

    }

    static public void main(String[] args) {
        System.out.println(new MagicHat("Gray Topper"));
        System.out.println(new MagicHat("Black Topper"));
        System.out.println(new MagicHat("Baseball Cap"));
        MagicHat oldHat = new MagicHat("Old hat");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\MagicHat.java

Gray Topper contains:
 MagicHat$Rabbit@275710fc  MagicHat$Rabbit@525f1e4e
MagicHat$Rabbit@75f9eccc MagicHat$Rabbit@52aa2946



Black Topper contains:

 MagicHat$Rabbit@4de5031f  MagicHat$Rabbit@67e2d983
MagicHat$Rabbit@5d47c63f
```

```
Baseball Cap contains:

 MagicHat$Rabbit@5ea434c8  MagicHat$Rabbit@3bbc39f8
MagicHat$Rabbit@4ae3c1cd   MagicHat$Rabbit@29f69090
MagicHat$Rabbit@568bf312
```

Q. Nested Class Demo



```
public class NestedClassDemo {
    public static void main(String args[]){
        int x[] = {3,2,1,5,6,9,7,8};
        Outer outob = new Outer(x);
        outob.analyze();

    }
}
class Outer{
    int nums[];
    Outer(int n[]){
        nums=n;
    }
    void analyze(){
        Inner inob = new Inner();
        System.out.println("Minimum : "+inob.min());
        System.out.println("Maximum: "+inob.max());
        System.out.println("Average : "+inob.avg());
    }
    class Inner{
        int min(){
            int m = nums[0];
            for(int i =1; i<nums.length; i++)
                if(nums[i] < m) m = nums[i];
                  return m;
        }
        int max(){
            int m = nums[0];
            for(int i =1; i<nums.length; i++)
                if(nums[i] > m) m = nums[i];
                  return m;
        }
        int avg(){
            int a = 0;
            for(int i =1; i<nums.length; i++)
              a += nums[i];

             return a/nums.length;
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\NestedClassDemo.java

Minimum : 1

Maximum: 9

Average : 4
```

Q. Interface

➔ Conversion.java

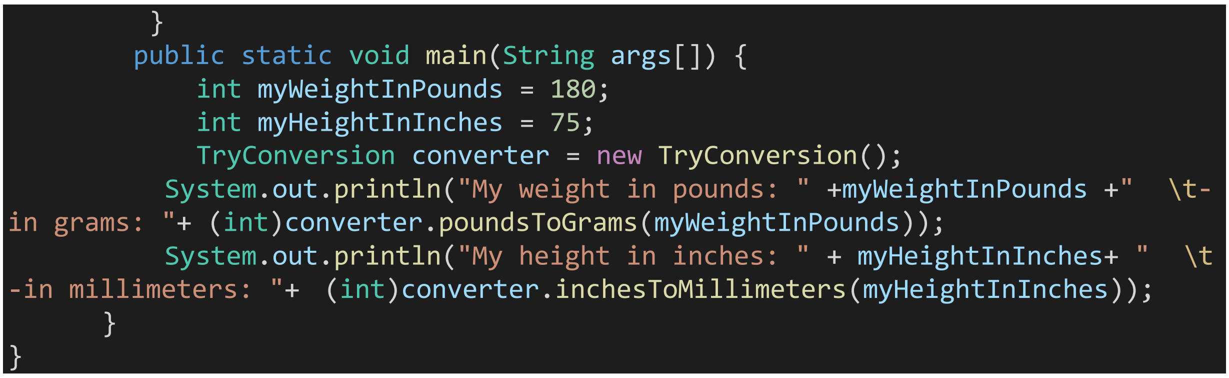
```
package conversions;
public interface Conversion {
    double inchesToMillimeters (double inches);
    double ouncesToGrams(double ounces);
    double poundsToGrams(double pounds);
    double hpToWatts(double hp);
    double wattsToHP(double watts);
}
```

➔ ConversionFactors.java

```
package conversions;
public interface ConversionFactors {
    double INCH_TO_MM = 25.4;
    double OUNCE_TO_GRAM = 28.349523125;
    double POUND_TO_GRAM = 453.5924;
    double HP_TO_WATT = 745.7;
    double WATT_TO_HP = 1.0/HP_TO_WATT;
}
```

➔ TryConversion.java

```
package conversions;
import static conversions.ConversionFactors.*; // Import static members
public class TryConversion implements Conversion {
    public double wattsToHP (double watts) {
            return watts*WATT_TO_HP;
       }
      public double hpToWatts (double hp) {
            return hp*HP_TO_WATT;
        }
      public double ouncesToGrams(double ounces) {
             return ounces*OUNCE_TO_GRAM;
        }
      public double poundsToGrams(double pounds) {
            return pounds*POUND_TO_GRAM;
        }
      public double inchesToMillimeters(double inches) {
           return inches*INCH_TO_MM;
```



```
         }
        public static void main(String args[]) {
            int myWeightInPounds = 180;
            int myHeightInInches = 75;
            TryConversion converter = new TryConversion();
          System.out.println("My weight in pounds: " +myWeightInPounds +"   \t-
in grams: "+ (int)converter.poundsToGrams(myWeightInPounds));
          System.out.println("My height in inches: " + myHeightInInches+ "   \t
-in millimeters: "+  (int)converter.inchesToMillimeters(myHeightInInches));
      }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\Conversion.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\ConversionFactors.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\TryConversion.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> java conversions.TryConversion

My weight in pounds: 180        -in grams: 81646

My height in inches: 75         -in millimeters: 1905
```

Q. Defining Interface

➔ RemoteControl.java

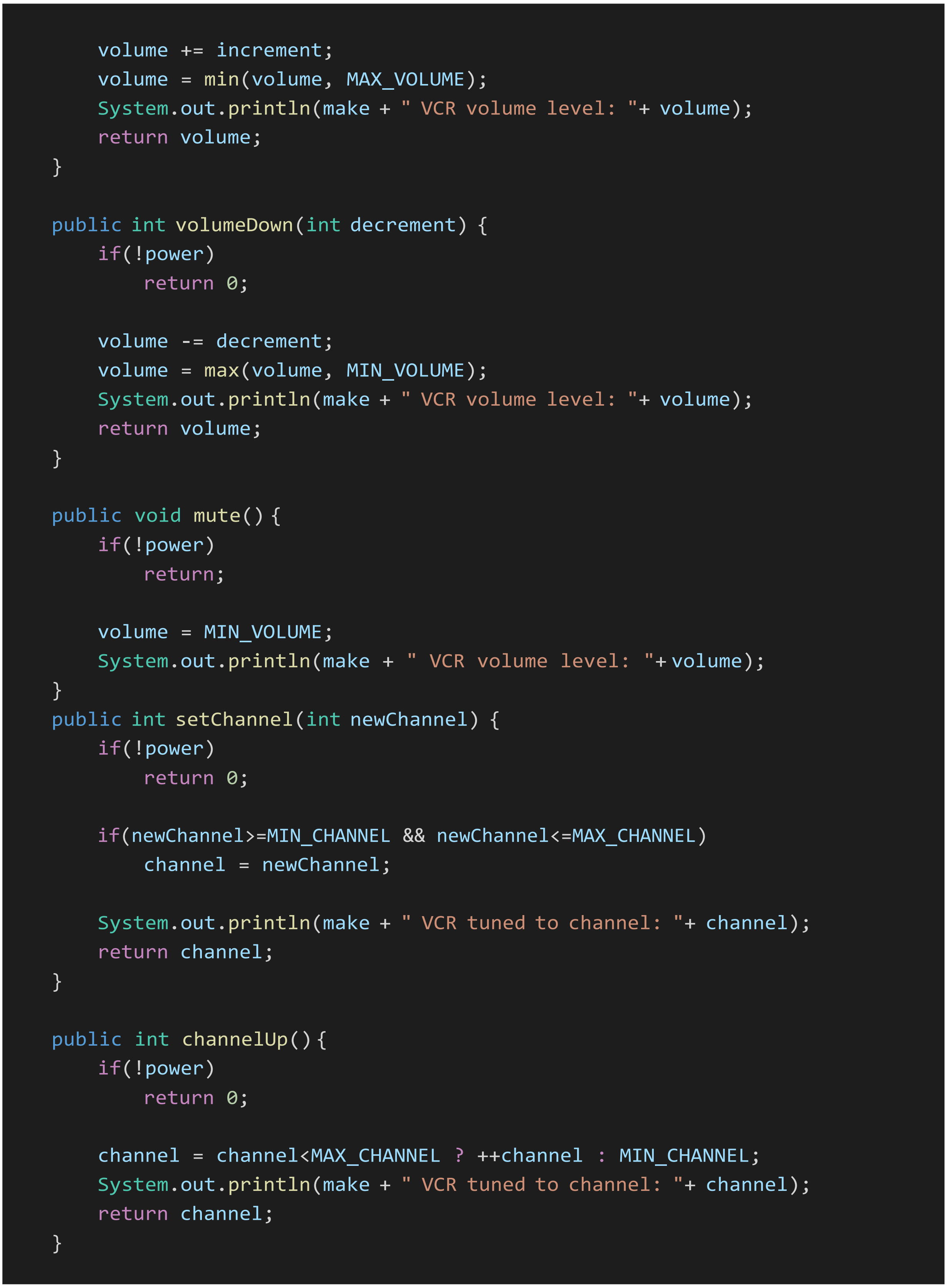
```java
package remote;
public interface RemoteControl {
        boolean powerOnOff(); // Returns new state, on = true
        int volumeUp(int increment); // Returns new volume level
        int volumeDown(int decrement); // Returns new volume level
        void mute(); // Mutes sound output
        int setChannel(int channel); // Set the channel number and return it
        int channelUp(); // Returns new channel number
        int channelDown(); // Returns new channel number
}
```

➔ TV.java

```java
package remote;

import static java.lang.Math.max;
import static java.lang.Math.min;

public class TV implements RemoteControl {

    public TV(String make, int screensize) {
        this.make = make;
        this.screensize = screensize;
    }

    public boolean powerOnOff() {
        power = !power;
        System.out.println(make + " "+ screensize + " inch TV power "+ (power ? " on.":"off."));
        return power;
    }

    public int volumeUp(int increment) {
        if(!power)
            return 0;

        volume += increment;
        volume = min(volume, MAX_VOLUME);
        System.out.println(make + " "+ screensize + " inch TV volume level: "+ vo
lume);
        return volume;
    }

    public int volumeDown(int decrement) {
        if(!power)
            return 0;

        volume -= decrement;
        volume = max(volume, MIN_VOLUME);
        System.out.println(make + " "+ screensize + " inch TV volume level: " + v
olume);
        return volume;
    }
    public void mute(){
        if(!power)
           return;

        volume = MIN_VOLUME;
        System.out.println(make + " "+ screensize + " inch TV volume level: "+ vo
lume);
    }

    public int setChannel(int newChannel) {
        if(!power)
            return 0;

        if(newChannel>=MIN_CHANNEL && newChannel<=MAX_CHANNEL)
            channel = newChannel;

        System.out.println(make + " "+ screensize + " inch TV tuned to channel: "
+ channel);
        return channel;
    }

    public int channelUp(){
        if(!power)
            return 0;

        channel = channel<MAX_CHANNEL ? ++channel : MIN_CHANNEL;
        System.out.println(make + " "+ screensize + " inch TV tuned to channel: "
+ channel);
        return channel;
    }

    public int channelDown(){
        if(!power)
            return 0;

        channel = channel>MIN_CHANNEL ? --channel : MAX_CHANNEL;
        System.out.println(make + " "+ screensize + " inch TV tuned to channel: "
+ channel);
        return channel;
    }
    private String make = null;
    private int screensize = 0;
    private boolean power = false;
    private int MIN_VOLUME = 0;
    private int MAX_VOLUME = 100;
    private int volume = MIN_VOLUME;
    private int MIN_CHANNEL = 0;
    private int MAX_CHANNEL = 999;
    private int channel = 0;
}
```

➔ **VCR.java**

```
package remote;
import static java.lang.Math.max;
import static java.lang.Math.min;
public class VCR implements RemoteControl {
    public VCR(String make) {
        this.make = make;
    }

    public boolean powerOnOff() {
        power = !power;
        System.out.println(make + " VCR power "+ (power ? "on.":"off."));
        return power;
    }

    public int volumeUp(int increment) {
        if(!power)
            return 0;
```



```
        volume += increment;
        volume = min(volume, MAX_VOLUME);
        System.out.println(make + " VCR volume level: "+ volume);
        return volume;
    }

    public int volumeDown(int decrement) {
        if(!power)
            return 0;

        volume -= decrement;
        volume = max(volume, MIN_VOLUME);
        System.out.println(make + " VCR volume level: "+ volume);
        return volume;
    }

    public void mute() {
        if(!power)
            return;

        volume = MIN_VOLUME;
        System.out.println(make + " VCR volume level: "+volume);
    }
    public int setChannel(int newChannel) {
        if(!power)
            return 0;

        if(newChannel>=MIN_CHANNEL && newChannel<=MAX_CHANNEL)
            channel = newChannel;

        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }

    public int channelUp() {
        if(!power)
            return 0;

        channel = channel<MAX_CHANNEL ? ++channel : MIN_CHANNEL;
        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }
    public int channelDown(){
        if(!power)
            return 0;

        channel = channel>MIN_CHANNEL ? --channel : MAX_CHANNEL;
        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }
    private String make = null;
    private boolean power = false;
    private int MIN_VOLUME = 0;
    private int MAX_VOLUME = 100;
    private int volume = MIN_VOLUME;
    private int MIN_CHANNEL = 0;
    private int MAX_CHANNEL = 99;
    private int channel = 0;
}
```

➔ TryRemoteControl.java

```
package remote;
import static java.lang.Math.random;
public class TryRemoteControl {
    public static void main(String args[]) {
        RemoteControl remote = null;
        for(int i = 0 ; i<5 ; i++) {
            if(random()<0.5)
                remote = new TV(random()<0.5 ? "Sony" : "Hitachi",random()<0.5 ?
32 : 28);
            else
                remote = new VCR(random()<0.5 ? "Panasonic": "JVC");
         remote.powerOnOff();
         remote.channelUp();
         remote.volumeUp(10);
      }
   }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\RemoteControl.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\TV.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\VCR.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\TryRemoteControl.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> java remote.TryRemoteControl

Panasonic VCR power on.

Panasonic VCR tuned to channel: 1

Panasonic VCR volume level: 10

JVC VCR power on.

JVC VCR tuned to channel: 1

JVC VCR volume level: 10

Hitachi 32 inch TV power on.

Hitachi 32 inch TV tuned to channel: 1

Hitachi 32 inch TV volume level: 10

JVC VCR power on.

JVC VCR tuned to channel: 1

JVC VCR volume level: 10

Hitachi 32 inch TV power on.

Hitachi 32 inch TV tuned to channel: 1

Hitachi 32 inch TV volume level: 10
```

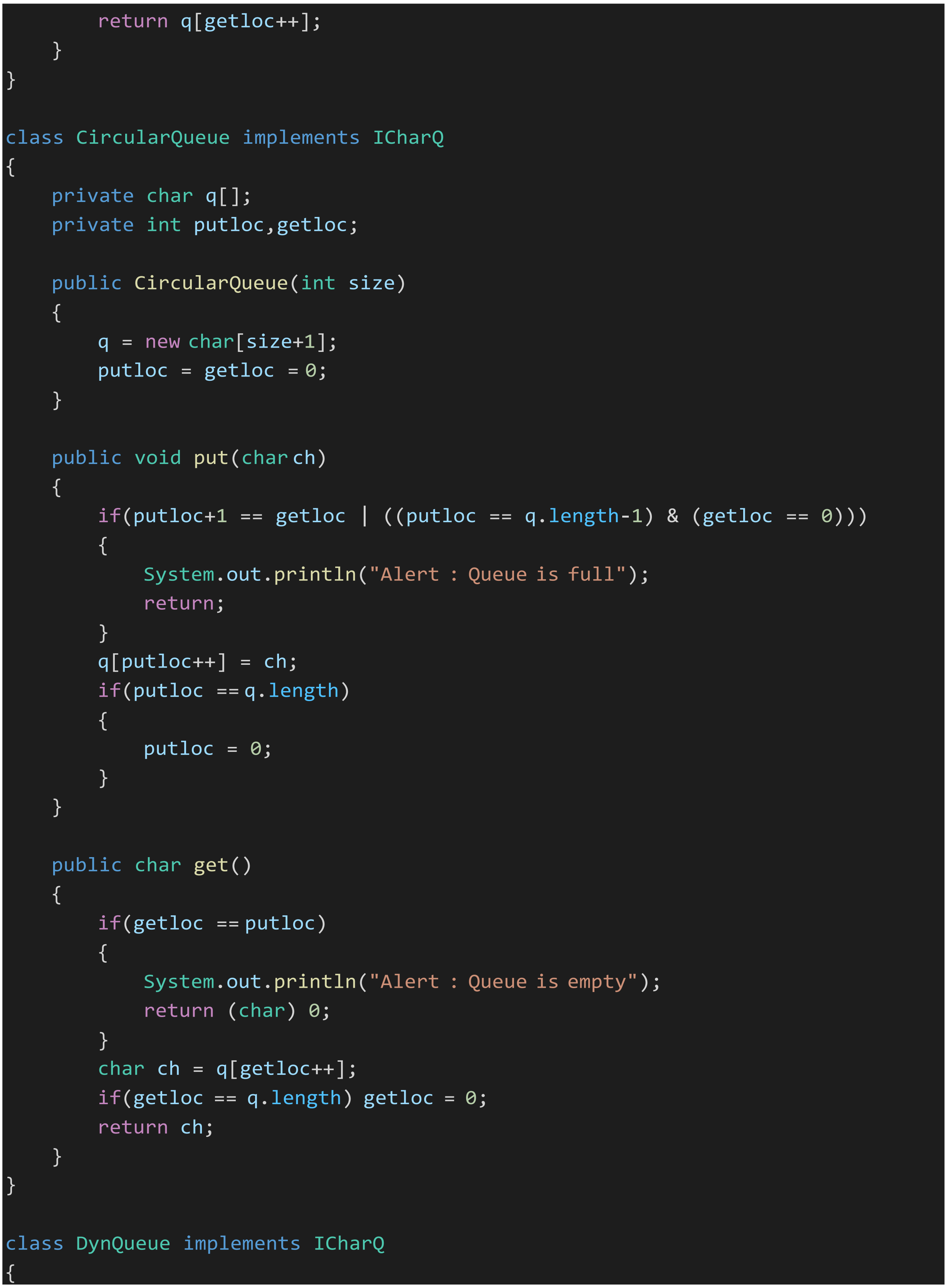
Q. Queue Interface

➔ ICharQ.java

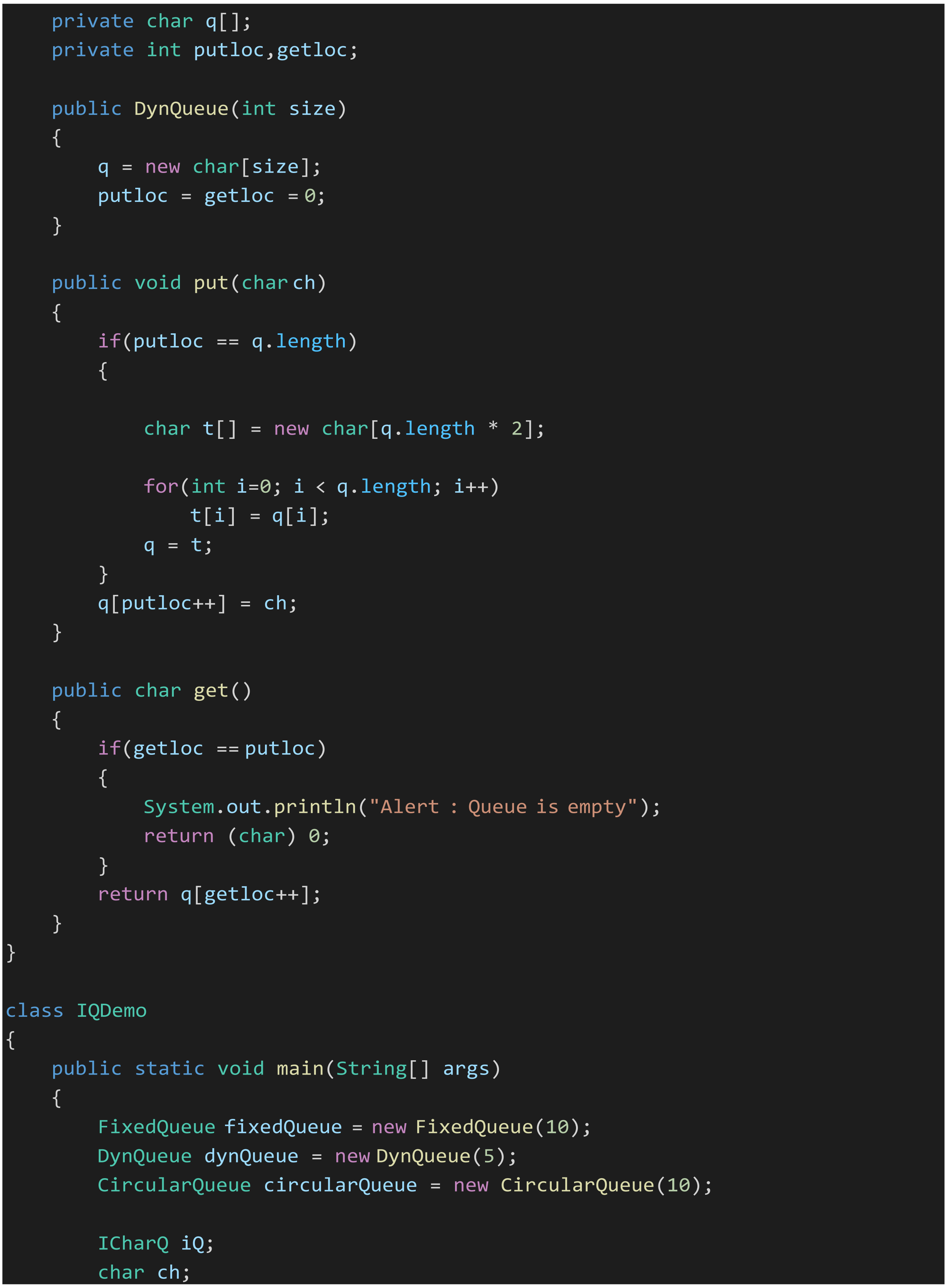
```
public interface ICharQ
{
    //Put a character into the queue
    void put(char ch);
    //Get a character from the queue
    char get();
}
```

➔ IQDemo.java

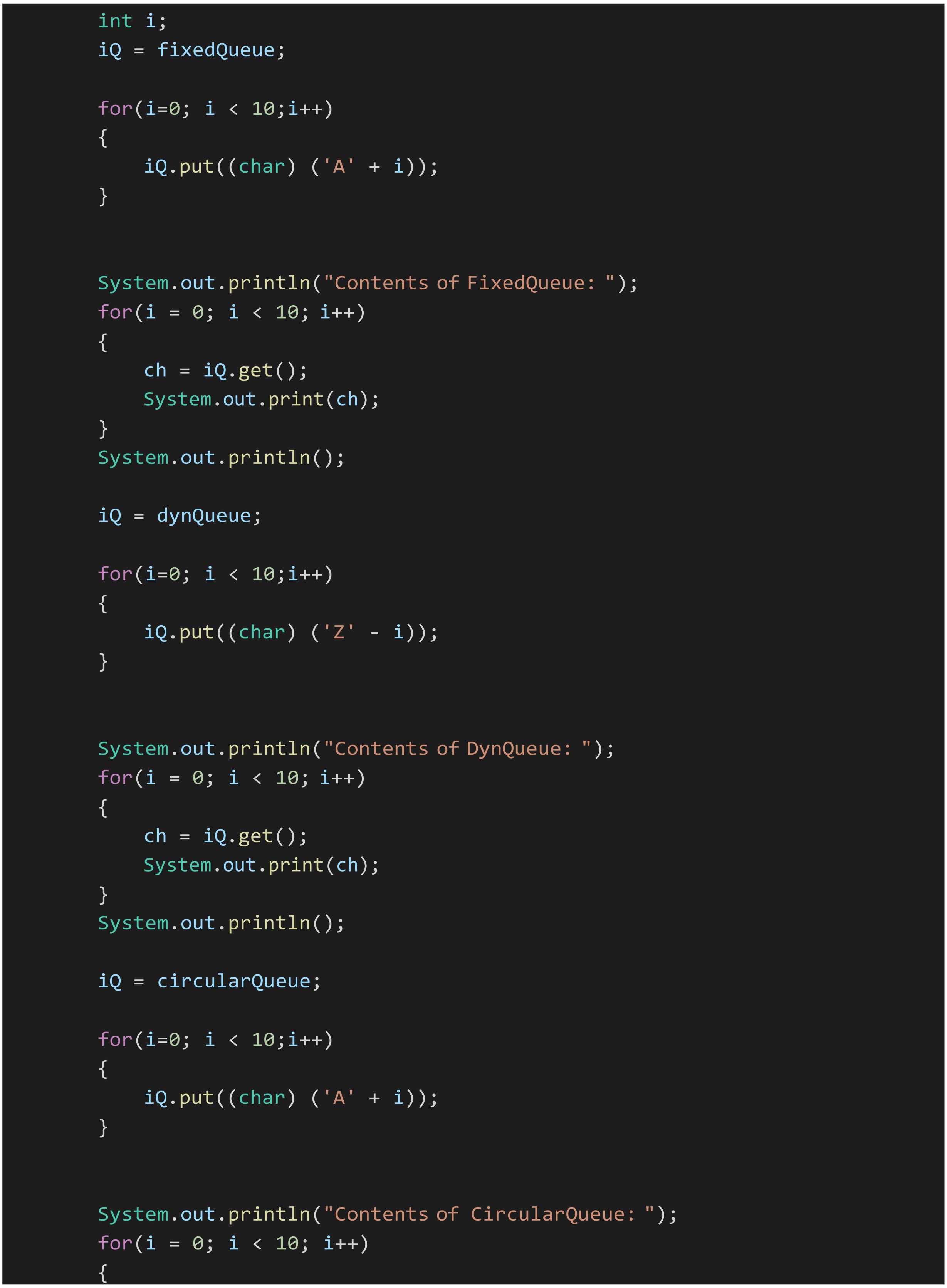
```
class FixedQueue implements ICharQ{
    private char q[];
    private int putloc,getloc;


    public FixedQueue(int size)
    {
        q = new char[size];
        putloc = getloc = 0;
    }


    public void put(char ch)
    {
        if(putloc == q.length)
        {
            System.out.println("Alert : Queue is full");
            return;
        }
        q[putloc++] = ch;
    }


    public char get()
    {
        if(getloc == putloc)
        {
            System.out.println("Alert : Queue is empty");
            return (char) 0;
        }
```



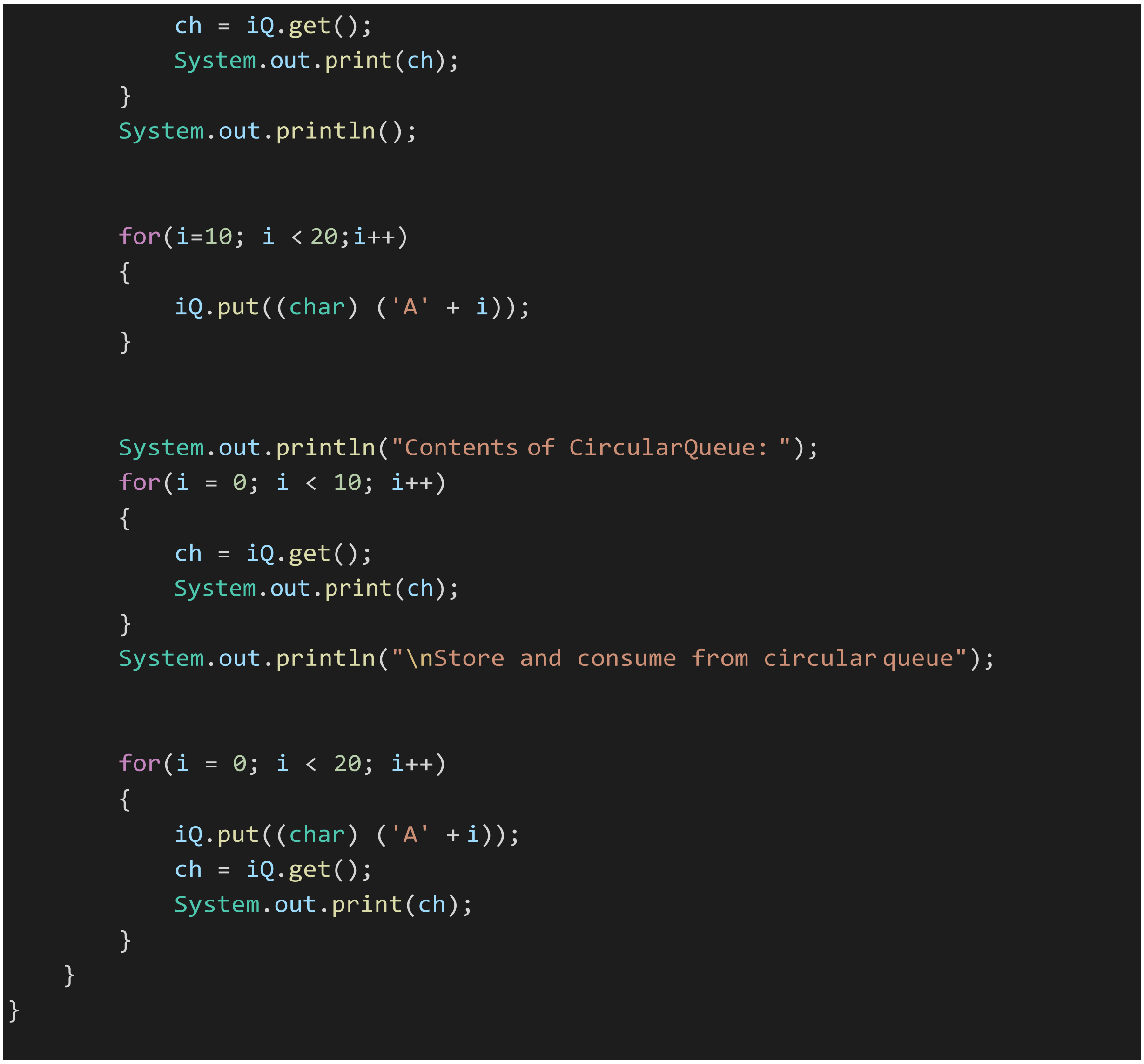
```
        return q[getloc++];
    }
}

class CircularQueue implements ICharQ
{
    private char q[];
    private int putloc,getloc;

    public CircularQueue(int size)
    {
        q = new char[size+1];
        putloc = getloc = 0;
    }

    public void put(char ch)
    {
        if(putloc+1 == getloc | ((putloc == q.length-1) & (getloc == 0)))
        {
            System.out.println("Alert : Queue is full");
            return;
        }
        q[putloc++] = ch;
        if(putloc ==q.length)
        {
            putloc = 0;
        }
    }

    public char get()
    {
        if(getloc ==putloc)
        {
            System.out.println("Alert : Queue is empty");
            return (char) 0;
        }
        char ch = q[getloc++];
        if(getloc == q.length) getloc = 0;
        return ch;
    }
}

class DynQueue implements ICharQ
{
```



```java
    private char q[];
    private int putloc,getloc;

    public DynQueue(int size)
    {
        q = new char[size];
        putloc = getloc =0;
    }

    public void put(char ch)
    {
        if(putloc == q.length)
        {

            char t[] = new char[q.length * 2];

            for(int i=0; i < q.length; i++)
                t[i] = q[i];
            q = t;
        }
        q[putloc++] = ch;
    }

    public char get()
    {
        if(getloc ==putloc)
        {
            System.out.println("Alert : Queue is empty");
            return (char) 0;
        }
        return q[getloc++];
    }
}

class IQDemo
{
    public static void main(String[] args)
    {
        FixedQueue fixedQueue = new FixedQueue(10);
        DynQueue dynQueue = new DynQueue(5);
        CircularQueue circularQueue = new CircularQueue(10);

        ICharQ iQ;
        char ch;
```



```
        int i;
        iQ = fixedQueue;

        for(i=0; i < 10;i++)
        {
            iQ.put((char) ('A' + i));
        }


        System.out.println("Contents of FixedQueue: ");
        for(i = 0; i < 10; i++)
        {
            ch = iQ.get();
            System.out.print(ch);
        }
        System.out.println();

        iQ = dynQueue;

        for(i=0; i < 10;i++)
        {
            iQ.put((char) ('Z' - i));
        }


        System.out.println("Contents of DynQueue: ");
        for(i = 0; i < 10; i++)
        {
            ch = iQ.get();
            System.out.print(ch);
        }
        System.out.println();

        iQ = circularQueue;

        for(i=0; i < 10;i++)
        {
            iQ.put((char) ('A' + i));
        }


        System.out.println("Contents of CircularQueue: ");
        for(i = 0; i < 10; i++)
        {
```



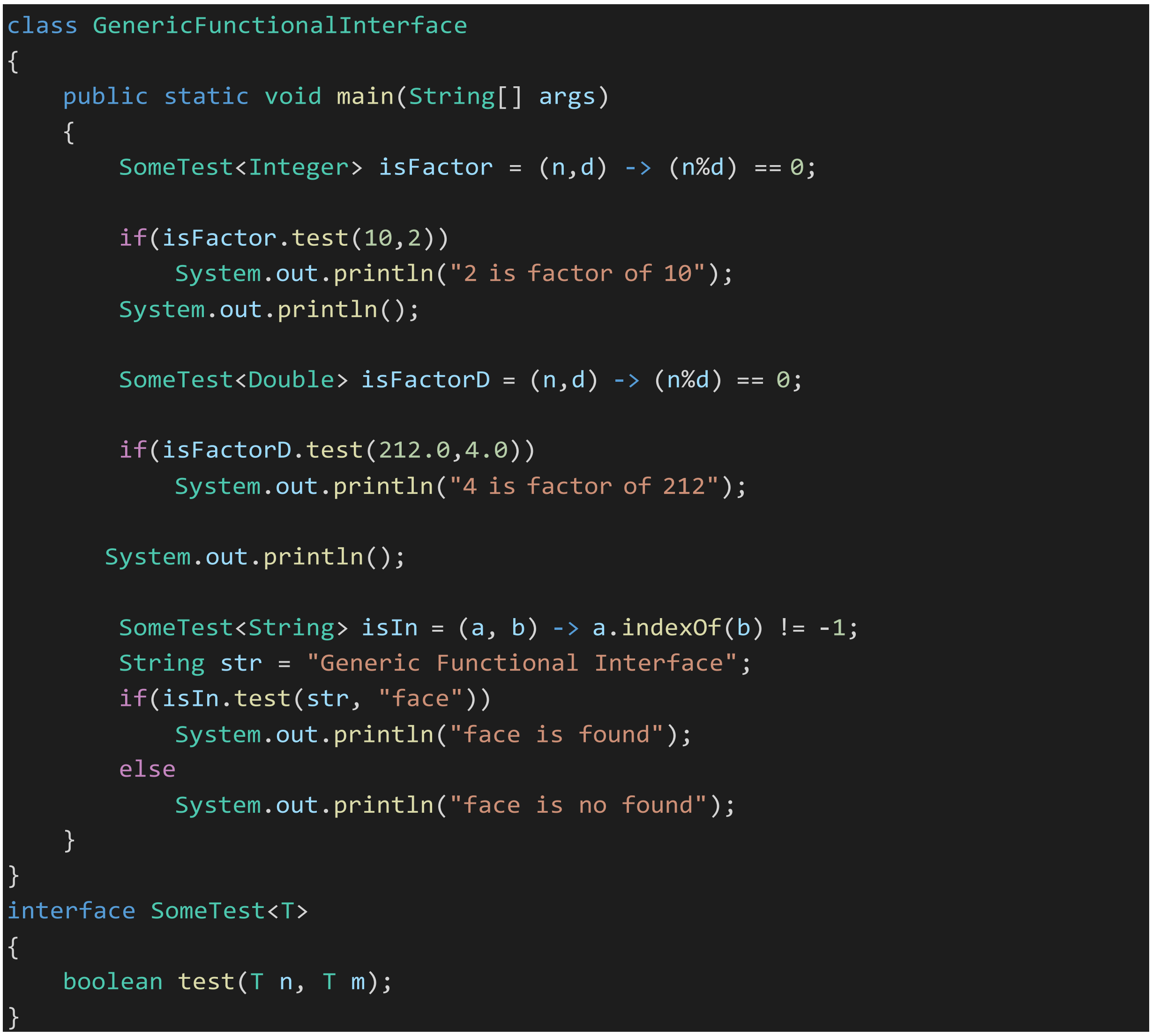
```java
            ch = iQ.get();
            System.out.print(ch);
        }
        System.out.println();


        for(i=10; i < 20;i++)
        {
            iQ.put((char) ('A' + i));
        }


        System.out.println("Contents of CircularQueue: ");
        for(i = 0; i < 10; i++)
        {
            ch = iQ.get();
            System.out.print(ch);
        }
        System.out.println("\nStore and consume from circular queue");


        for(i = 0; i < 20; i++)
        {
            iQ.put((char) ('A' +i));
            ch = iQ.get();
            System.out.print(ch);
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> javac .\ICharQ.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac .\IQDemo.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> java IQDemo
```

```
Contents of FixedQueue:
ABCDEFGHIJ
Contents of DynQueue:
ZYXWVUTSRQ
Contents of CircularQueue:
ABCDEFGHIJ
Contents of CircularQueue:
KLMNOPQRST
Store and consume from circular queue
ABCDEFGHIJKLMNOPQRST
```

Q. Generic Functional Interface

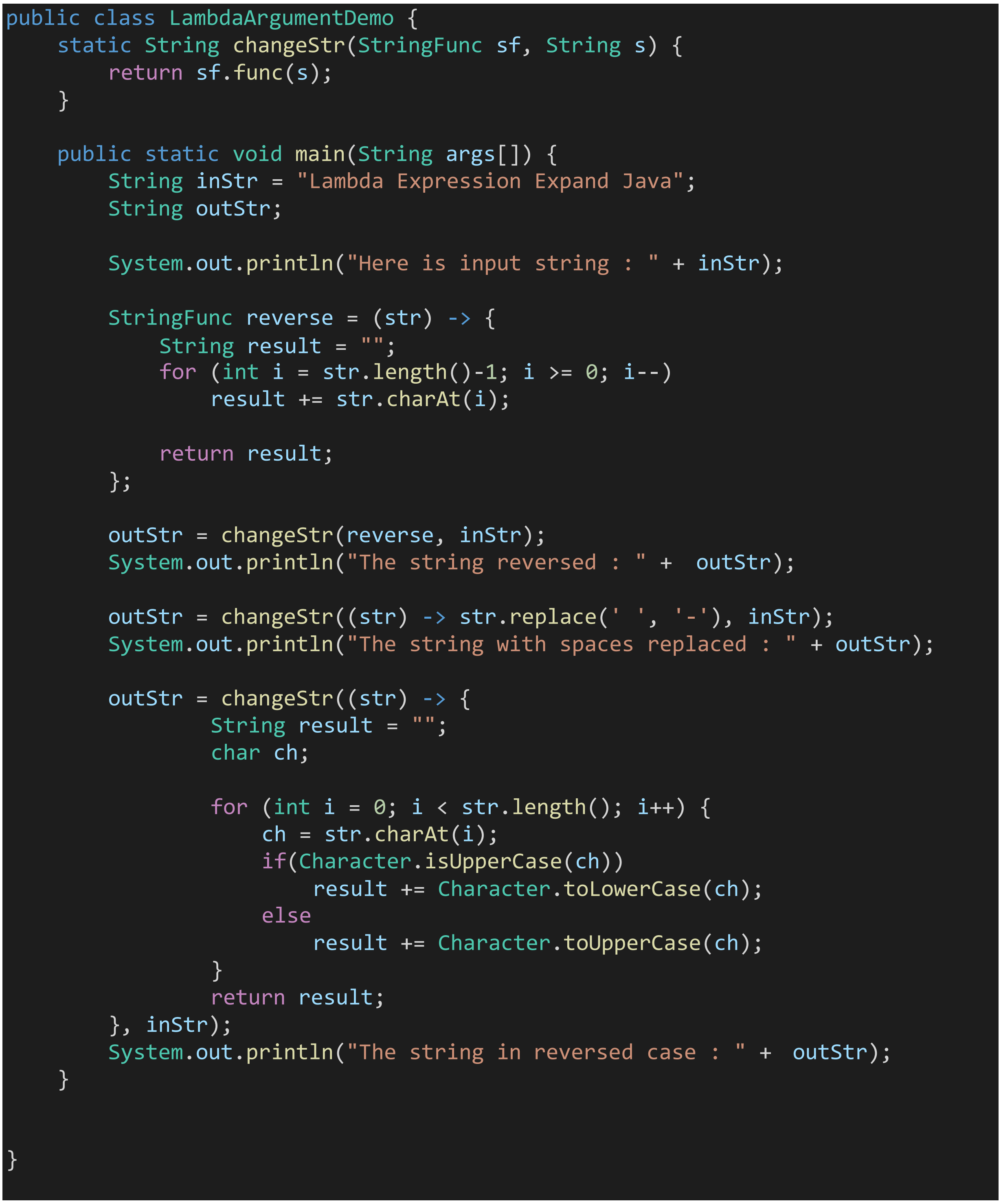


```java
class GenericFunctionalInterface
{
    public static void main(String[] args)
    {
        SomeTest<Integer> isFactor = (n,d) -> (n%d) == 0;

        if(isFactor.test(10,2))
            System.out.println("2 is factor of 10");
        System.out.println();

        SomeTest<Double> isFactorD = (n,d) -> (n%d) == 0;

        if(isFactorD.test(212.0,4.0))
            System.out.println("4 is factor of 212");

       System.out.println();

        SomeTest<String> isIn = (a, b) -> a.indexOf(b) != -1;
        String str = "Generic Functional Interface";
        if(isIn.test(str, "face"))
            System.out.println("face is found");
        else
            System.out.println("face is no found");
    }
}
interface SomeTest<T>
{
    boolean test(T n, T m);
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\GenericFunctionalInterface.java

2 is factor of 10

4 is factor of 212

face is found
```

Q. Lambda Argument Demo



```java
public class LambdaArgumentDemo {
    static String changeStr(StringFunc sf, String s) {
        return sf.func(s);
    }

    public static void main(String args[]) {
        String inStr = "Lambda Expression Expand Java";
        String outStr;

        System.out.println("Here is input string : " + inStr);

        StringFunc reverse = (str) -> {
            String result = "";
            for (int i = str.length()-1; i >= 0; i--)
                result += str.charAt(i);

            return result;
        };

        outStr = changeStr(reverse, inStr);
        System.out.println("The string reversed : " +  outStr);

        outStr = changeStr((str) -> str.replace(' ', '-'), inStr);
        System.out.println("The string with spaces replaced : " + outStr);

        outStr = changeStr((str) -> {
                String result = "";
                char ch;

                for (int i = 0; i < str.length(); i++) {
                    ch = str.charAt(i);
                    if(Character.isUpperCase(ch))
                        result += Character.toLowerCase(ch);
                    else
                        result += Character.toUpperCase(ch);
                }
                return result;
        }, inStr);
        System.out.println("The string in reversed case : " +  outStr);
    }


}
```

```java
interface StringFunc {
    String func(String  str);
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\LambdaArgumentDemo.java

Here is input string : Lambda Expression Expand Java

The string reversed : avaJ dnapxE noisserpxE  adbmaL

The string with spaces replaced : Lambda-Expression-Expand-Java

The string in reversed case : lAMBDA eXPRESSION eXPAND jAVA
```

Q. Lambda Exception Demo

```
import java.io.*;

  class LambdaExceptionDemo{
    public static void main(String args[])
    {
        double[] values = {1.0,2.0,3.0,4.0};
        MyIOAction myIO = (rdr) -> {
              int ch = rdr.read();
              return true;
        };
    }
}
interface MyIOAction{
    boolean ioAction(Reader rdr) throws IOException;
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\LambdaExceptionDemo.java

PS D:\MCA\MCA SEM 3\JAVA\Practice>
```

Q. Package ExtInterface

➔ ExtInterface.java

```java
package extInterface;
public interface ExtInterface {
    public void method1();
    public void method2();

}
```

➔ TestExtInterface.java

```java
package extInterface;

import java.util.Scanner;

class TestExtInterface implements ExtInterface{
    public void method1() {
        System.out.println("Implementation of Method1");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number to find the Square root in java:");
        double square = sc.nextDouble();
        double squareRoot = Math.sqrt(square);
        System.out.printf("Square root of number: %f is -
> %f",square,squareRoot);

    }
    public void method2() {
        System.out.println("Implementation of method2");
    }
    public static void main(String[] args) {
        ExtInterface obj  = new TestExtInterface();
        obj.method1();
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice\35> javac –d . .\ExtInterface.java

PS D:\MCA\MCA SEM 3\JAVA\Practice\35> javac –d . .\TestExtInterface.java

PS D:\MCA\MCA SEM 3\JAVA\Practice\35> java extInterface.TestExtInterface

Implementation of Method1

Enter number to find the Square root in java:

9

Square root of number: 9.000000 is -> 3.000000
```

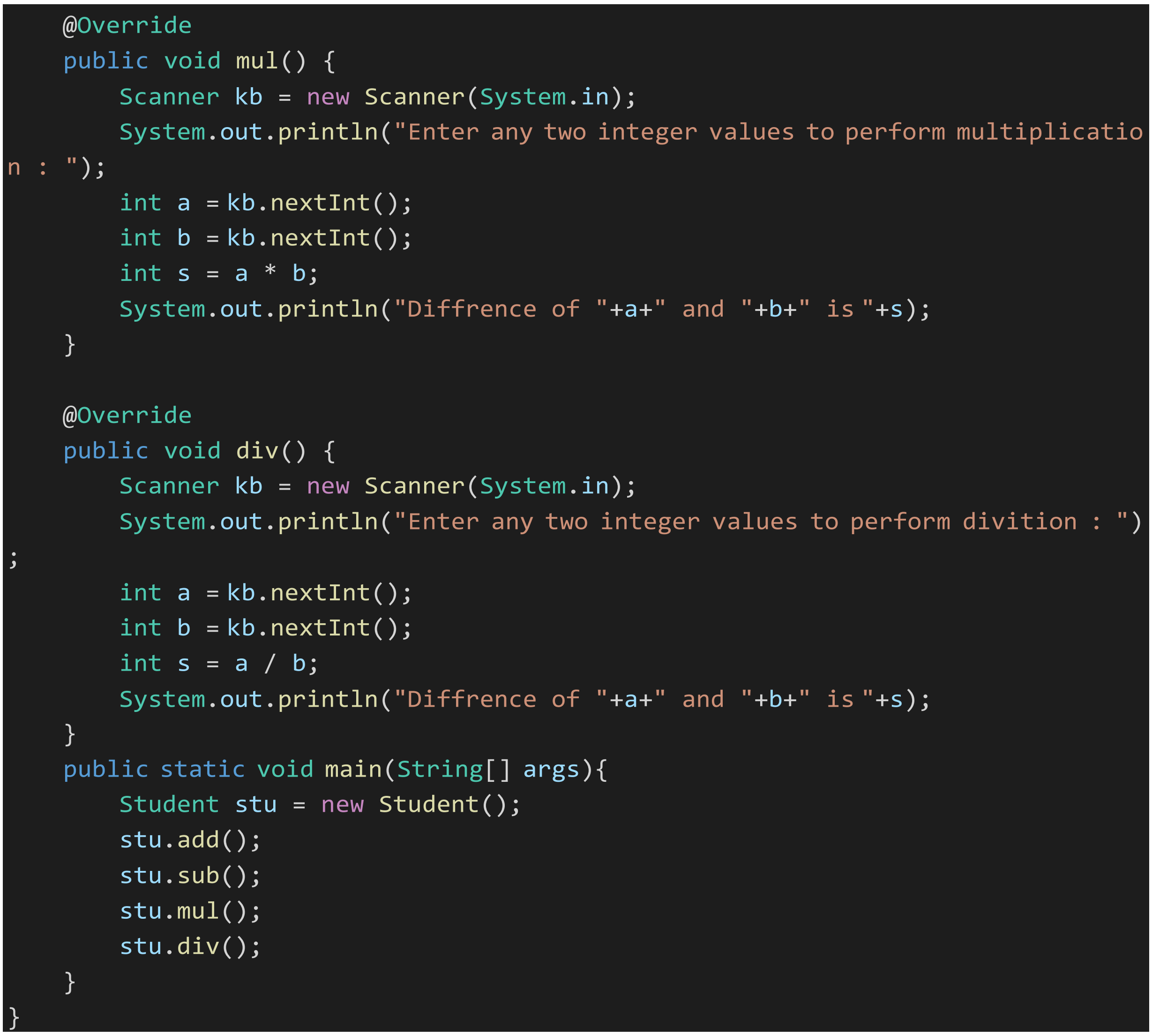
Q. Package BasicOperation

➔ Math.java

```
package basicoperation;
public interface Math {
    public void add();
    public void sub();
    public void mul();
    public void div();
}
```

➔ Student.java

```
package basicoperation;
import java.util.Scanner;
public class Student implements math{
    @Override
    public void add() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform addition : ")
;
        int a = kb.nextInt();
        int b = kb.nextInt();
        int s = a + b;
        System.out.println("Diffrence of "+a+" and "+b+" is "+s);
    }

    @Override
    public void sub() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform subtraction :
 ");
        int a = kb.nextInt();
        int b = kb.nextInt();
        int s = a - b;
        System.out.println("Diffrence of "+a+" and "+b+" is "+s);
    }
```



```
    @Override
    public void mul() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform multiplicatio
n : ");
        int a = kb.nextInt();
        int b = kb.nextInt();
        int s = a * b;
        System.out.println("Diffrence of "+a+" and "+b+" is "+s);
    }

    @Override
    public void div() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform divition : ")
;
        int a = kb.nextInt();
        int b = kb.nextInt();
        int s = a / b;
        System.out.println("Diffrence of "+a+" and "+b+" is "+s);
    }
    public static void main(String[] args){
        Student stu = new Student();
        stu.add();
        stu.sub();
        stu.mul();
        stu.div();
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java basicoperation.Student

Enter any two integer values to perform addition :

10 20

Diffrence of 10 and 20 is 30

Enter any two integer values to perform subtraction :

30 20
```

```
Diffrence of 30 and 20 is 10

Enter any two integer values to perform multiplication :

2

30

Diffrence of 2 and 30 is 60

Enter any two integer values to perform divition :

40 2

Diffrence of 40 and 2 is 20
```

Q. Package mca

➔ Solution.java

```java
package mca;
public interface Solution{
    public void Hello();
    public void Welcome();
}
```

➔ classA.java

```java
package mca;
public class classA implements Solution {
    public void Hello(){
        System.out.println("Hello World");
    }
    public void Welcome(){
        System.out.println("Welcome  MCAIII");
    }

    public static void main(String[] args) {
        classA test = new classA();
        test.Hello();
        test.Welcome();
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java mca.classA

Hello World

Welcome MCAIII
```

Q. Package book

➔ Books.java

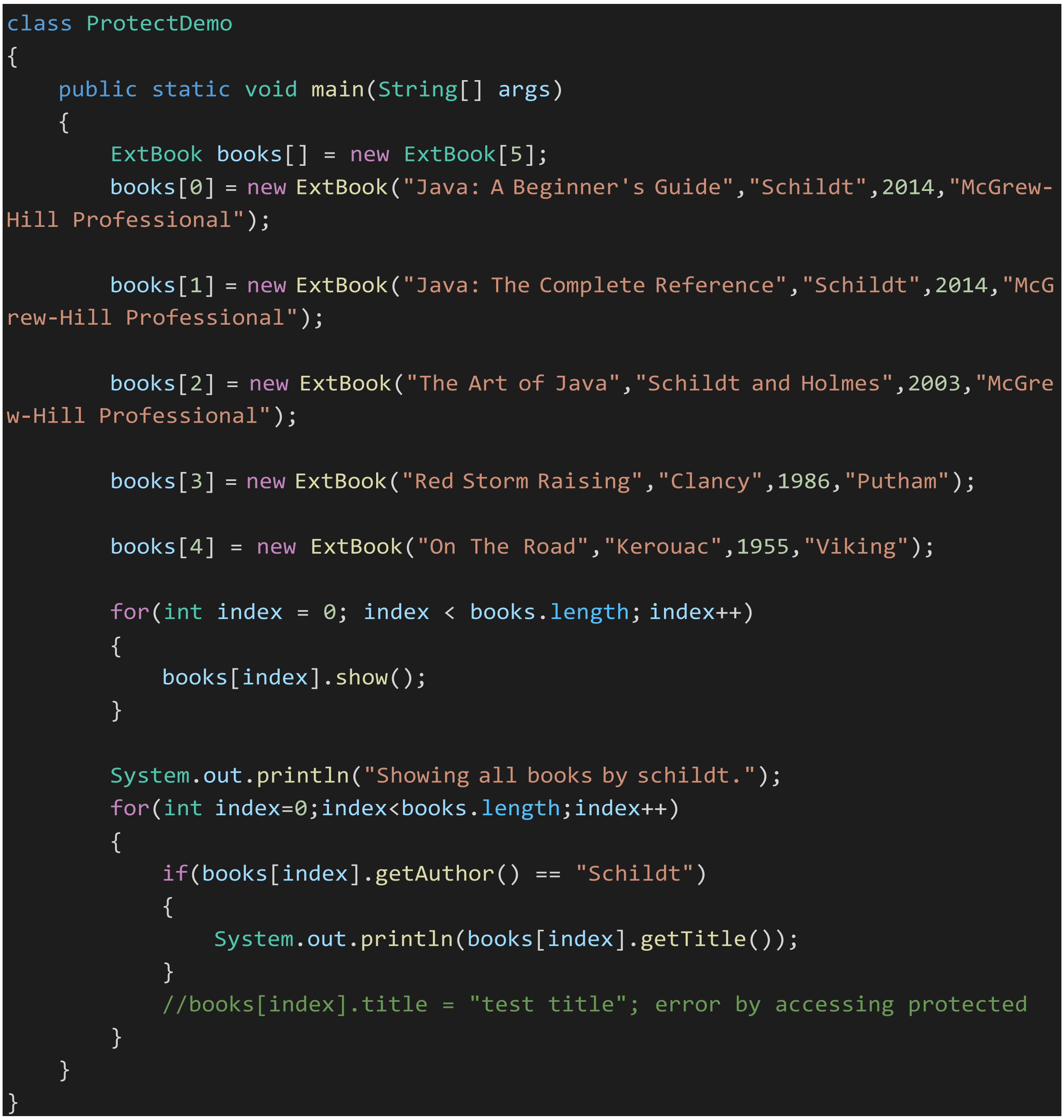
```
package bookpack;

public class Books{
    //these are protected members
    protected String title;
    protected String author;
    protected int pubDate;

    public Books(String t, String a, int d)
    {
        title = t;
        author = a;
        pubDate = d;
    }

    public void show()
    {
        System.out.println(title);
        System.out.println(author);
        System.out.println(pubDate);
        System.out.println();
    }
}
```

➔ UseBook.java

```
package bookpackext;
import bookpack.*;

class UseBook {
    public static void main(String args[])
    {
        Books book[] = new Books[5];
        book[0] = new Books("Java: A Beginner's Guide","Schildt",2014);

        book[1] = new Books("Java: The Complete Reference","Schildt",2014);

        book[2] = new Books("The Art of Java","Schildt and Holmes",2003);

        book[3] = new Books("Red Storm Raising","Clancy",1986);

        book[4] = new Books("On The Road","Kerouac",1955);

        for(int index = 0; index < book.length;index++) book[index].show();
    }
}
```

➔ **ProtectDemo.java**

```
package bookpackext;

class ExtBook extends bookpack.Books{
    private String publisher;
    public ExtBook(String t,String a, int d,String p)
    {
        super(t,a,d);
        publisher = p;
    }

    public void show()
    {
        super.show();
        System.out.println(publisher);
        System.out.println();
    }

    public String getPublisher() { return publisher; }
    public void setPublisher(String p) { publisher = p; }

    public String getTitle() { return title; }
    public void setTitle(String t) { title = t; }

    public String getAuthor() { return author; }
    public void setAuthor(String a) { author = a; }

    public int getPubDate() { return pubDate; }
    public void setTitle(int d) { pubDate = d; }

}
```



```java
class ProtectDemo
{
    public static void main(String[] args)
    {
        ExtBook books[] = new ExtBook[5];
        books[0] = new ExtBook("Java: A Beginner's Guide","Schildt",2014,"McGrew-Hill Professional");

        books[1] = new ExtBook("Java: The Complete Reference","Schildt",2014,"McGrew-Hill Professional");

        books[2] = new ExtBook("The Art of Java","Schildt and Holmes",2003,"McGrew-Hill Professional");

        books[3] = new ExtBook("Red Storm Raising","Clancy",1986,"Putham");

        books[4] = new ExtBook("On The Road","Kerouac",1955,"Viking");

        for(int index = 0; index < books.length; index++)
        {
            books[index].show();
        }

        System.out.println("Showing all books by schildt.");
        for(int index=0;index<books.length;index++)
        {
            if(books[index].getAuthor() == "Schildt")
            {
                System.out.println(books[index].getTitle());
            }
            //books[index].title = "test title"; error by accessing protected
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice>  javac -d . .\Books.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d .  .\UseBook.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\ProtectDemo.java
PS D:\MCA\MCA SEM 3\JAVA\Practice> java bookpackext.ProtectDemo
Java: A Beginner's Guide
Schildt
2014

McGrew-Hill Professional

Java: The Complete Reference
Schildt
2014

McGrew-Hill Professional

The Art of Java
Schildt and Holmes
2003

McGrew-Hill Professional

Red Storm Raising
Clancy
1986
```

```
Putham

On The Road
Kerouac
1955

Viking

Showing all books by schildt.
Java: A Beginner's Guide
Java: The Complete Reference
```

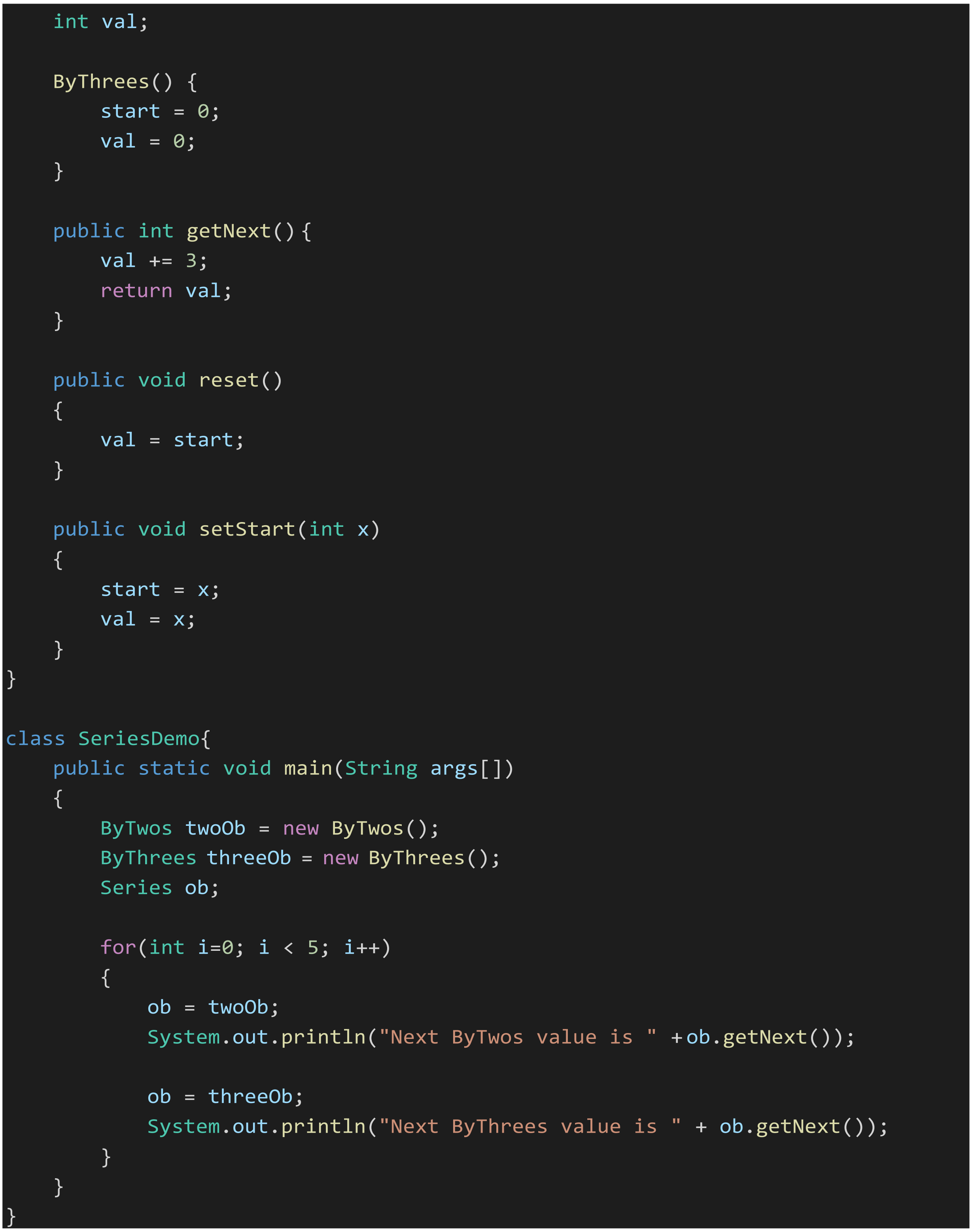
Q. Series Program

➔ Series.java

```
public interface Series
{
    int getNext();
    void reset();
    void setStart(int x);
}
```

➔ SeriesDemo.java

```
class ByTwos implements Series
{
    int start;
    int val;

    ByTwos() {
        start = 0;
        val = 0;
    }

    public int getNext(){
        val += 2;
        return val;
    }

    public void reset()
    {
        val = start;
    }

    public void setStart(int x)
    {
        start = x;
        val = x;
    }
}

class ByThrees implements Series
{
    int start;
```

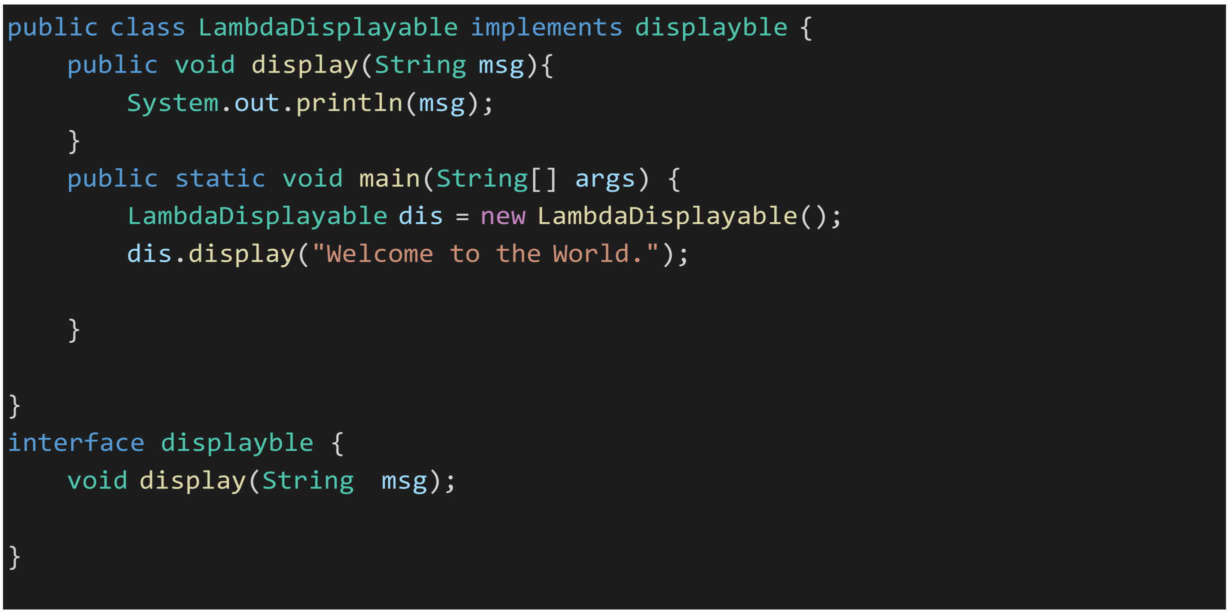


```
    int val;

    ByThrees() {
        start = 0;
        val = 0;
    }

    public int getNext(){
        val += 3;
        return val;
    }

    public void reset()
    {
        val = start;
    }

    public void setStart(int x)
    {
        start = x;
        val = x;
    }
}

class SeriesDemo{
    public static void main(String args[])
    {
        ByTwos twoOb = new ByTwos();
        ByThrees threeOb = new ByThrees();
        Series ob;

        for(int i=0; i < 5; i++)
        {
            ob = twoOb;
            System.out.println("Next ByTwos value is " +ob.getNext());

            ob = threeOb;
            System.out.println("Next ByThrees value is " + ob.getNext());
        }
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\Series.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> javac -d . .\SeriesDemo.java

PS D:\MCA\MCA SEM 3\JAVA\Practice> java SeriesDemo

Next ByTwos value is 2

Next ByThrees value is 3

Next ByTwos value is 4

Next ByThrees value is 6

Next ByTwos value is 6

Next ByThrees value is 9

Next ByTwos value is 8

Next ByThrees value is 12

Next ByTwos value is 10

Next ByThrees value is 15
```

Q. Lambda Displayable



```
public class LambdaDisplayable implements displayble {
    public void display(String msg){
        System.out.println(msg);
    }
    public static void main(String[] args) {
        LambdaDisplayable dis = new LambdaDisplayable();
        dis.display("Welcome to the World.");

    }

}
interface displayble {
    void display(String  msg);

}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\LambdaDisplayable.java

Welcome to the World.
```

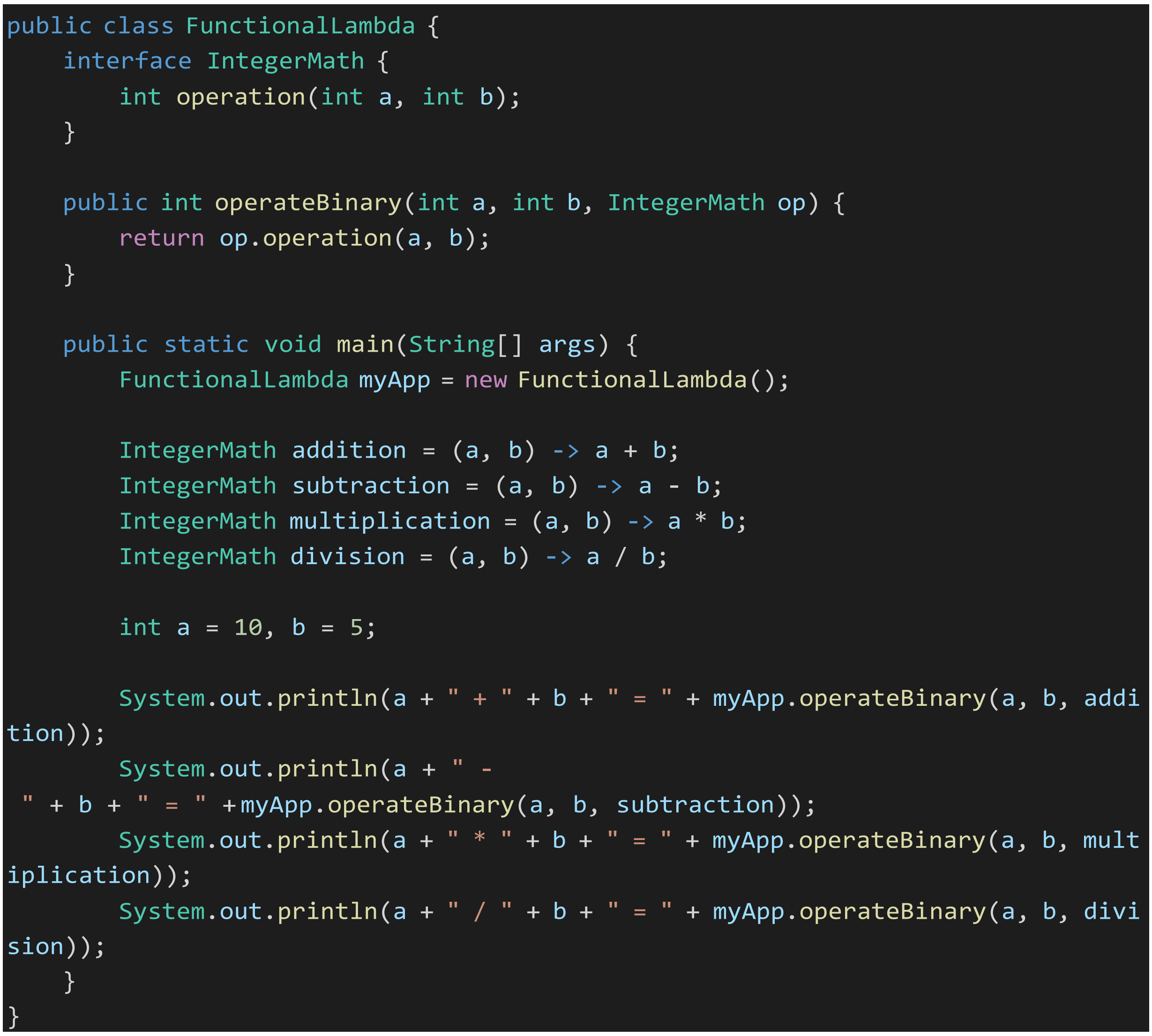
Q. Lambda with functional Interface

```java
public class LambdaFuntionalInterface {
    public static void main(String[] args)
    {
        FuncInterface fobj  = (int x) -> System.out.println(2*x);
        fobj.abstractFun(12);
    }
}

interface FuncInterface{
    void abstractFun(int x);

    default void normalFun()
    {
        System.out.println("Hello");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\LambdaFuntionalInterface.java

24
```

Q. Calculator Using Lambda



```
public class FunctionalLambda {
    interface IntegerMath {
        int operation(int a, int b);
    }

    public int operateBinary(int a, int b, IntegerMath op) {
        return op.operation(a, b);
    }

    public static void main(String[] args) {
        FunctionalLambda myApp = new FunctionalLambda();

        IntegerMath addition = (a, b) -> a + b;
        IntegerMath subtraction = (a, b) -> a - b;
        IntegerMath multiplication = (a, b) -> a * b;
        IntegerMath division = (a, b) -> a / b;

        int a = 10, b = 5;

        System.out.println(a + " + " + b + " = " + myApp.operateBinary(a, b, addi
tion));
        System.out.println(a + " -
 " + b + " = " +myApp.operateBinary(a, b, subtraction));
        System.out.println(a + " * " + b + " = " + myApp.operateBinary(a, b, mult
iplication));
        System.out.println(a + " / " + b + " = " + myApp.operateBinary(a, b, divi
sion));
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\FunctionalLambda.java

10 + 5 = 15

10 - 5 = 5

10 * 5 = 50

10 / 5 = 2
```

Q. Implement Uber using interface, Anonymous class and Lambda Expression

➔ Uber1.java

```java
public class Uber1 {
    public static void main(String[] args) {
        Cab cab = new UberX1();
        cab.bookCab();
    }
}

@FunctionalInterface
interface Cab {
    void bookCab();
}

class UberX1 implements Cab {
    public void bookCab() {
        System.out.println("UberX Booked!! Arriving Soon !!");
    }
}
```

Output :

```
PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Uber1.java

UberX Booked!! Arriving Soon !!
```

➔ Uber2.java

```java
public class Uber2 {
    public static void main(String[] args) {
        Cab cab = new Cab() {

            @Override
            public void bookCab() {
                System.out.println("UberX Booked!! Arriving Soon !!");
            }
        };

        cab.bookCab();
    }
}

@FunctionalInterface
interface Cab {
    void bookCab();
}
```

Output :

PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Uber2.java

UberX Booked!! Arriving Soon !!

➔ **Uber3.java**

```
public class Uber3 {
    public static void main(String[] args) {
        Cab cab = () -> System.out.println("UberX Booked!! Arriving Soon !!");
        cab.bookCab();
    }
}

@FunctionalInterface
interface Cab {
    void bookCab();
}
```

Output :

PS D:\MCA\MCA SEM 3\JAVA\Practice> java .\Uber3.java

UberX Booked!! Arriving Soon !!