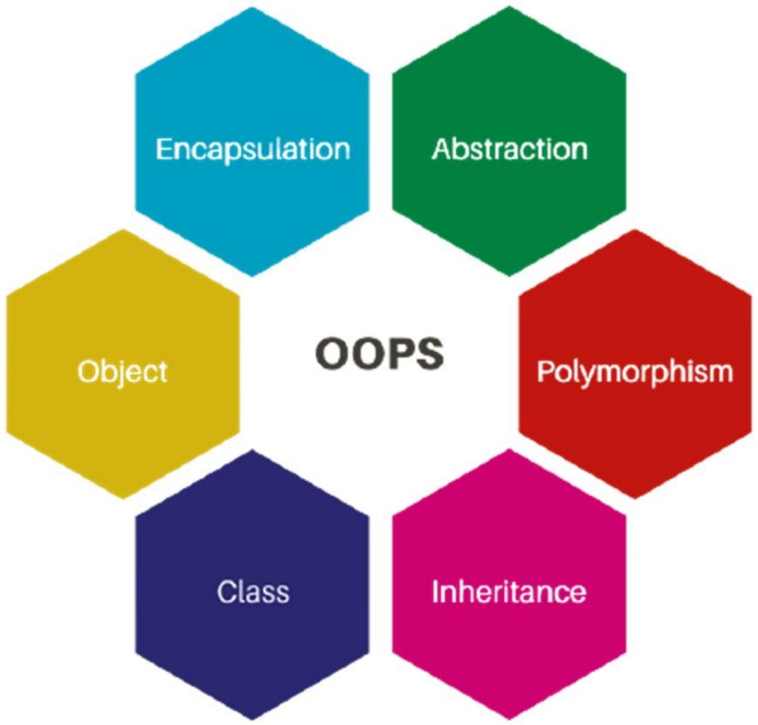


ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Core Java - String

ABESIT COLLEGE OF ENGINEERING

CRC-Training

Strings in Java

- In Java, a string is an object. It is not a primitive type.
- The String class is used to create and store immutable strings.
  - Immutable objects are objects that don't change once created.
- Class StringBuilder creates objects that store flexible and changeable strings.
  - StringBuilder → Create mutable String
  - We'll learn this later on in the course.

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

The String class

- Part of java.lang package
- Near about 50 inbuilt methods
- Once you build a String object, it is fixed – it cannot be changed.
- You can assign a String reference variable to a new string, discarding the old one
- When we create a string in java, it actually creates an object of type String.

CRC-Training

Different Ways to Create String

- **Using string literal**

```
String str = "abc";
```

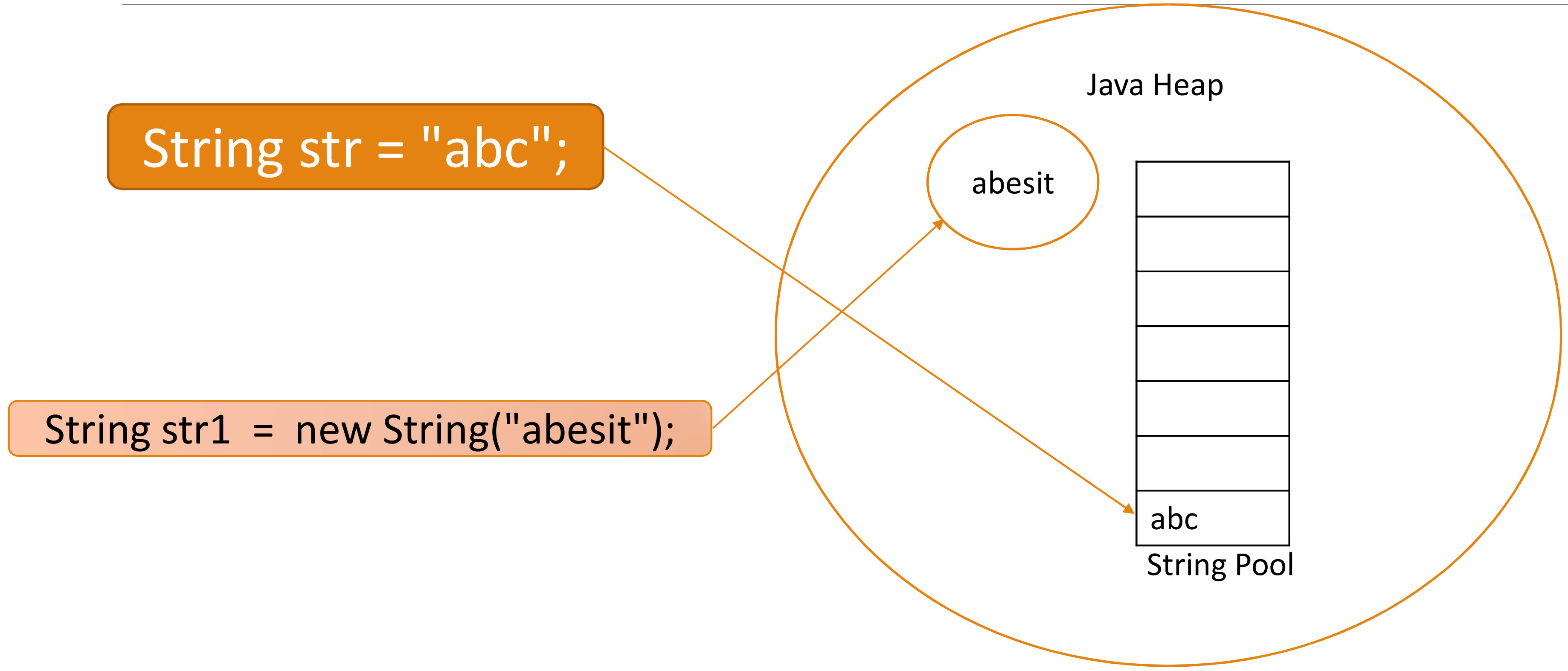
- **Using new keyword**

```
String str1  =  new String("abc");
```

```
char[] arr = {'a','b','e','s','i','t'};
String str = new String(arr);
```

Memory Representation

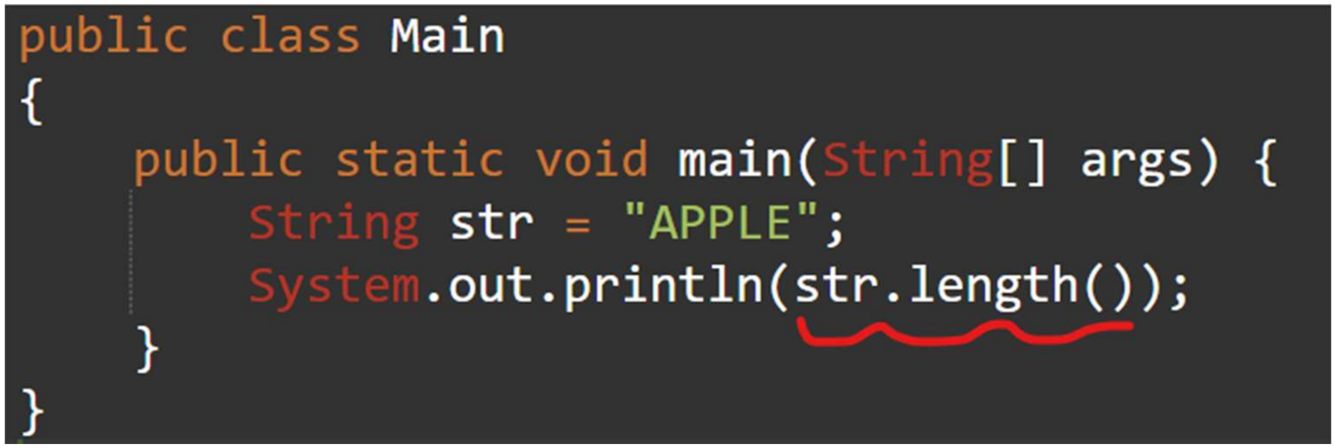
CRC-Training



Inbuilt methods - length()

CRC-Training

length() method finds the length of a string.



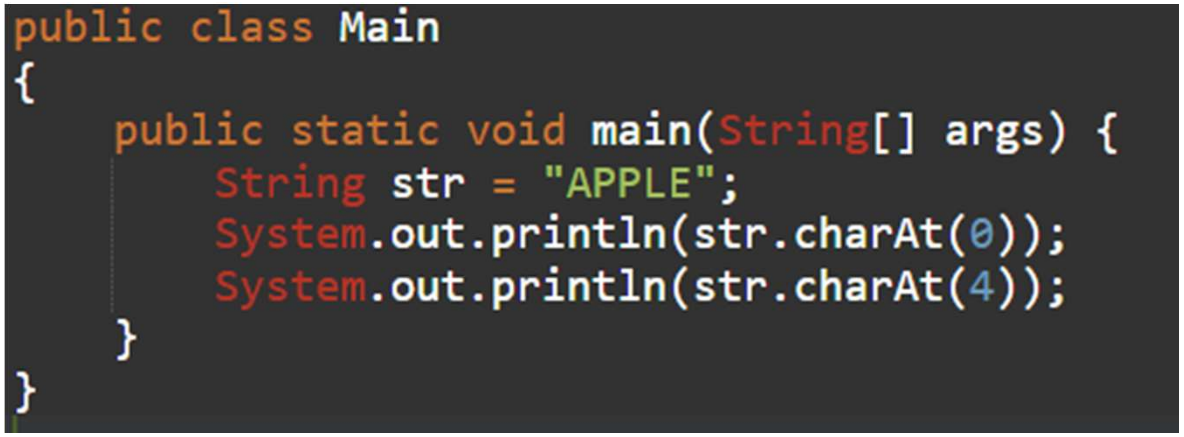
```
public class Main
{
    public static void main(String[] args) {
        String str = "APPLE";
        System.out.println(str.length());
    }
}
```

Output
5

CRC-Training

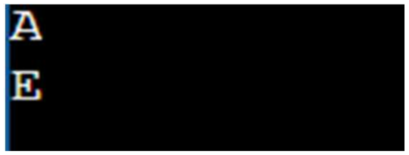
Inbuilt methods - charAt()

charAt() method returns a char value at the given index number.



```
public class Main
{
    public static void main(String[] args) {
        String str = "APPLE";
        System.out.println(str.charAt(0));
        System.out.println(str.charAt(4));
    }
}
```

Output



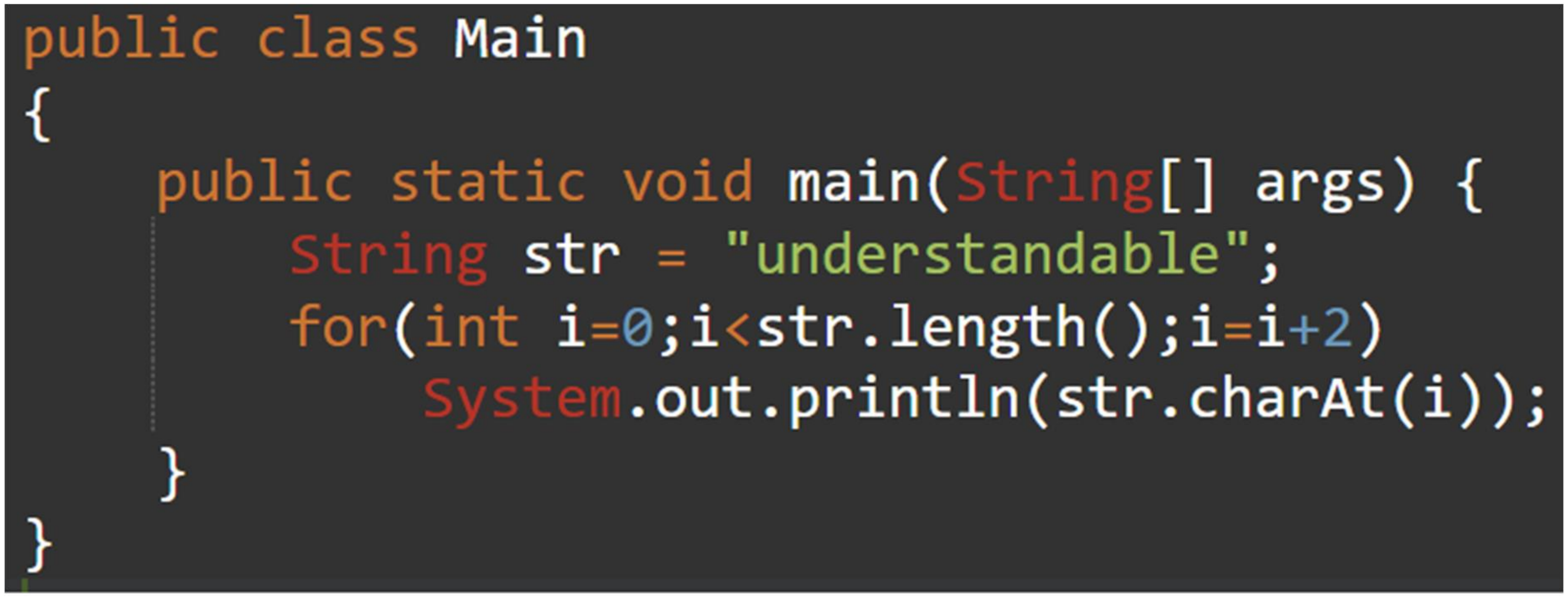
```
A
E
```

Note - It returns **StringIndexOutOfBoundsException**, if the given index number **is greater than or equal** to this string length or a negative number.

CRC-Training

Inbuilt methods - charAt()

Write a Program to print alternate char from a given String.(i.e index – 0,2,4,6,8,10,……….)



```
public class Main
{
    public static void main(String[] args) {
        String str = "understandable";
        for(int i=0;i<str.length();i=i+2)
            System.out.println(str.charAt(i));
    }
}
```

Inbuilt methods - substring()

substring() method returns a part of the string.

Syntax –

Return string starting from given index to end of string

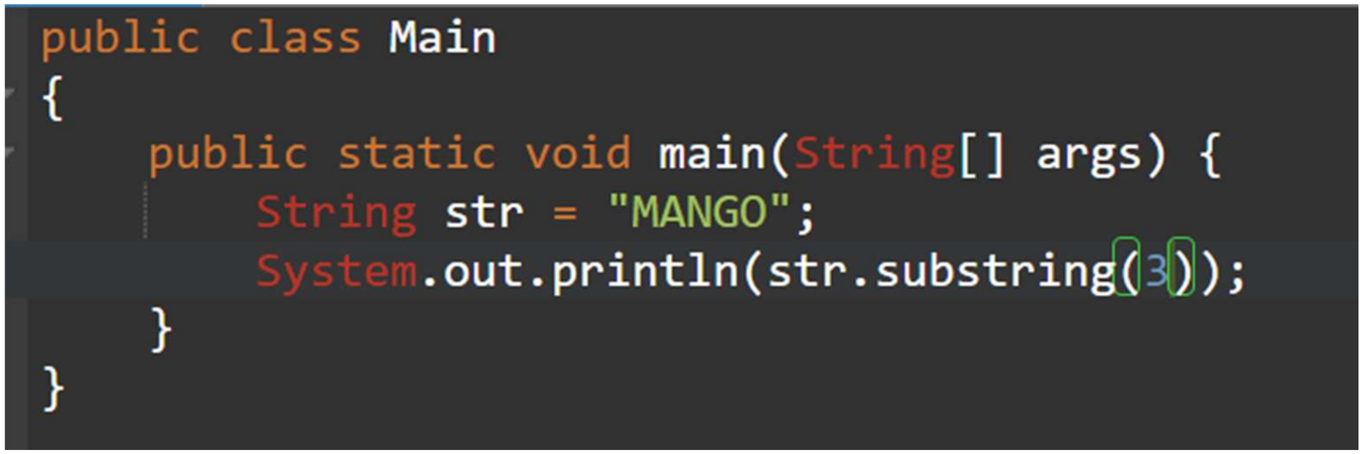
```
str.substring(start-index)
```

Start-index – inclusive
End-index - exclusive

```
str.substring(start-index , end-Index)
```

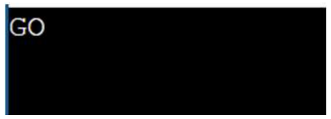
CRC-Training

Inbuilt methods - substring()



```
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.substring(3));
    }
}
```

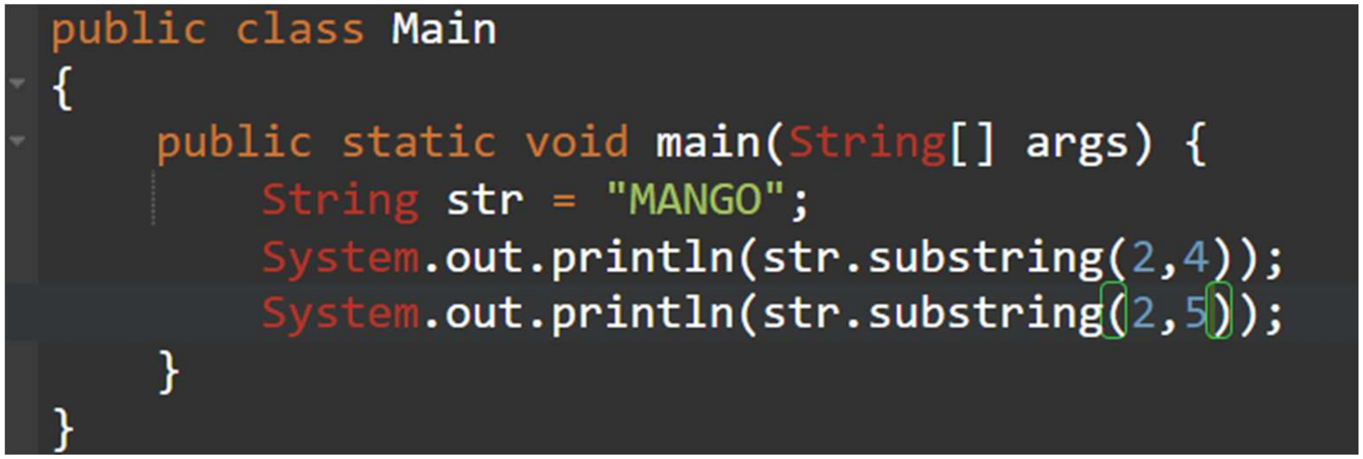
Output -



```
GO
```

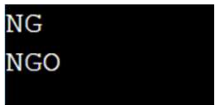
CRC-Training

Inbuilt methods - substring()



```java
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.substring(2,4));
        System.out.println(str.substring(2,5));
    }
}
```

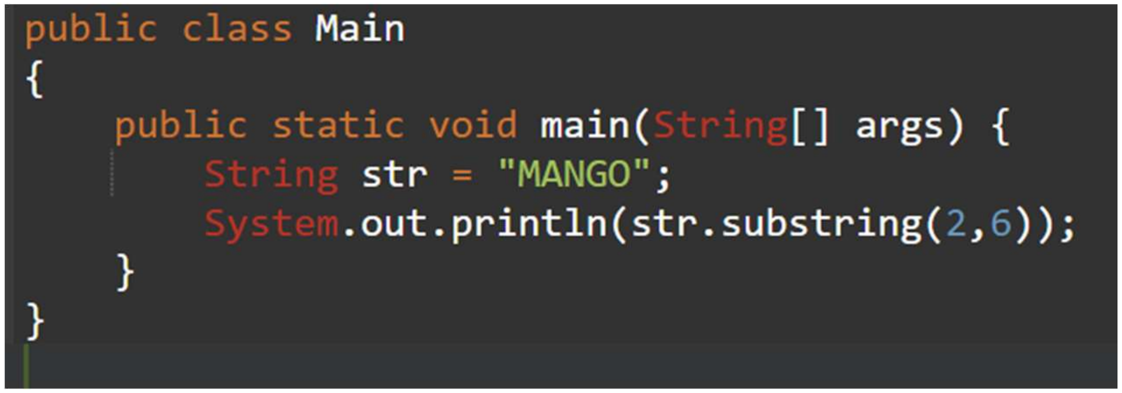
Output -



```
NG
NGO
```

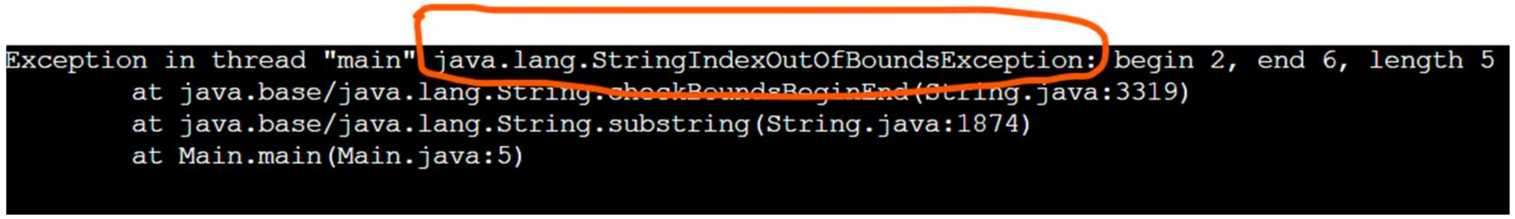
Inbuilt methods - substring()

What if end index is greater than length of string?



```
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.substring(2,6));
    }
}
```

StringIndexOutOfBoundsException



```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: begin 2, end 6, length 5
        at java.base/java.lang.String.checkBoundsBeginEnd(String.java:3319)
        at java.base/java.lang.String.substring(String.java:1874)
        at Main.main(Main.java:5)
```

CRC-Training

Inbuilt methods - substring()

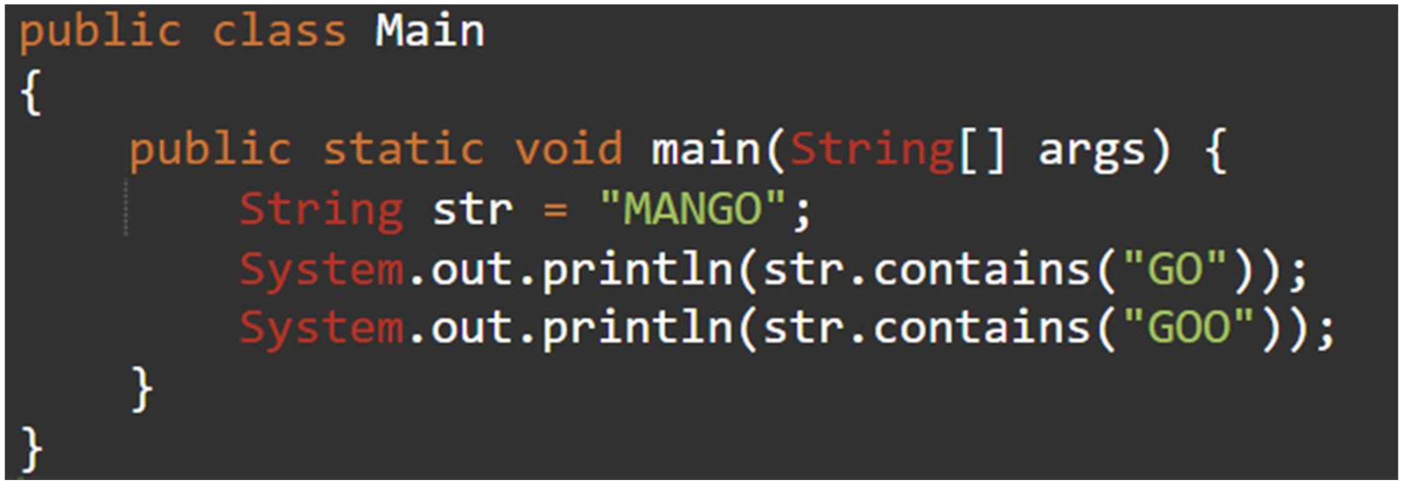
Write a Java program which takes two index as input and print substring starting from given index upto given index.

CRC-Training

Inbuilt methods - contains()

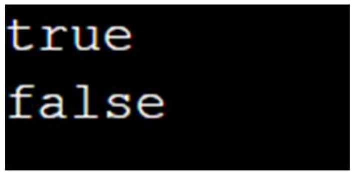
contains() method searches the sequence of characters in this string.

It returns **true if the sequence of char values is found** in this string otherwise **returns false**.



```java
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.contains("GO"));
        System.out.println(str.contains("GOO"));
    }
}
```

Output -

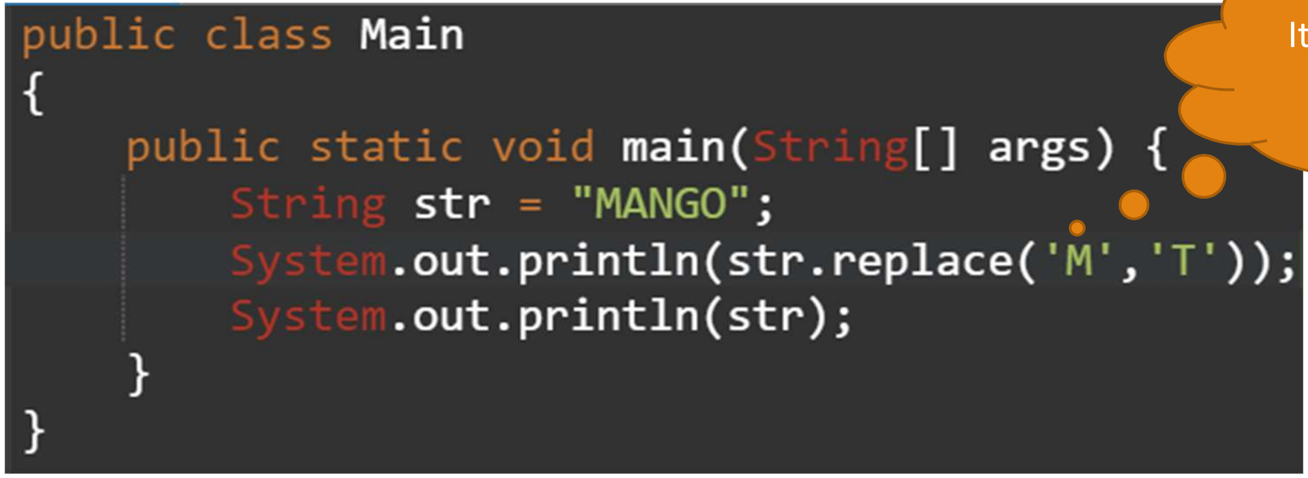


```
true
false
```

CRC-Training

Inbuilt methods - replace()

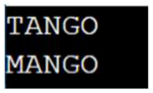
replace() method returns a string replacing all the **old char or CharSequence** to **new char or CharSequence.**



```
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.replace('M','T'));
        System.out.println(str);
    }
}
```

It return a new String after replacing

Output -

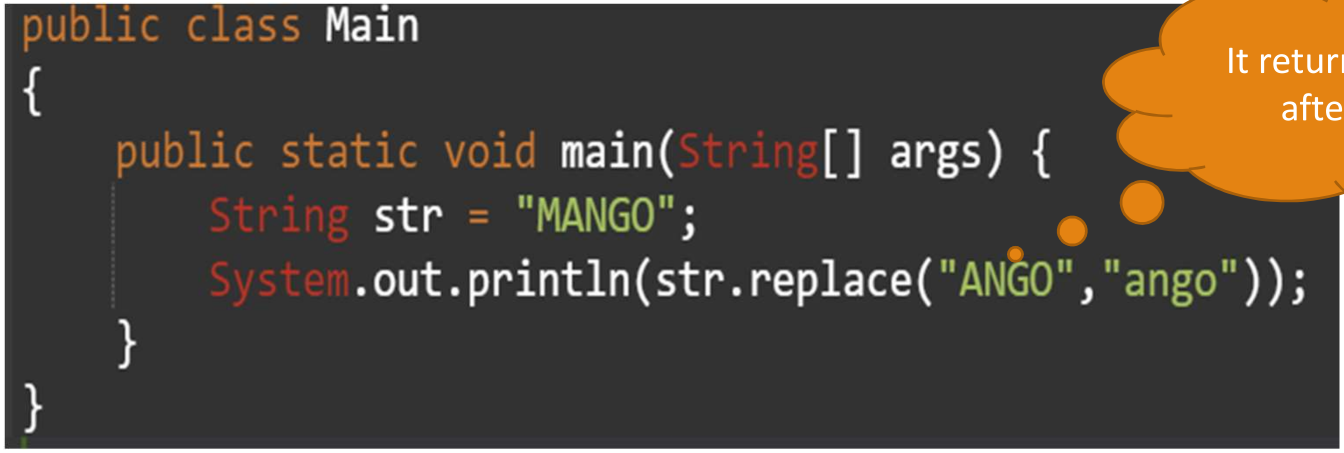


```
TANGO
MANGO
```

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Inbuilt methods - replace()

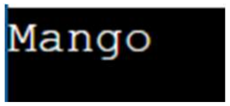
replace() method returns a string replacing all the **old char or CharSequence** to **new char or CharSequence.**



```
public class Main
{
    public static void main(String[] args) {
        String str = "MANGO";
        System.out.println(str.replace("ANGO","ango"));
    }
}
```

It return a new String after replacing

Output -



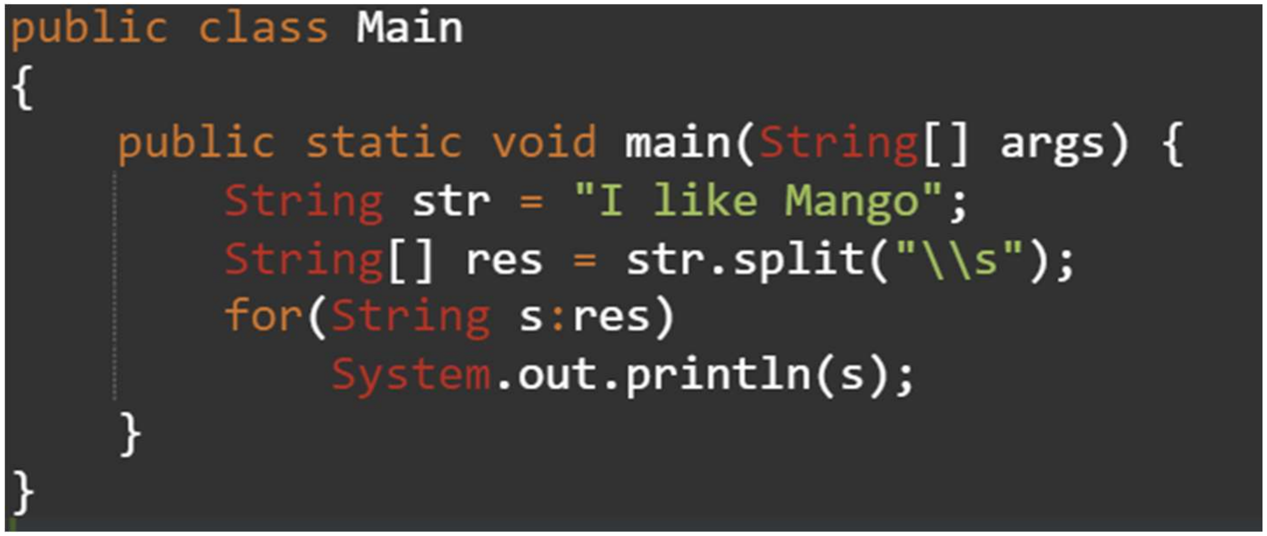
```
Mango
```

CRC-Training

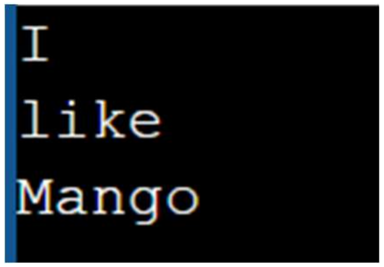
Inbuilt methods - split()

split() method splits this string against given regular expression and returns a char array.

splits the string based on whitespace



```
public class Main
{
    public static void main(String[] args) {
        String str = "I like Mango";
        String[] res = str.split("\\s");
        for(String s:res)
            System.out.println(s);
    }
}
```



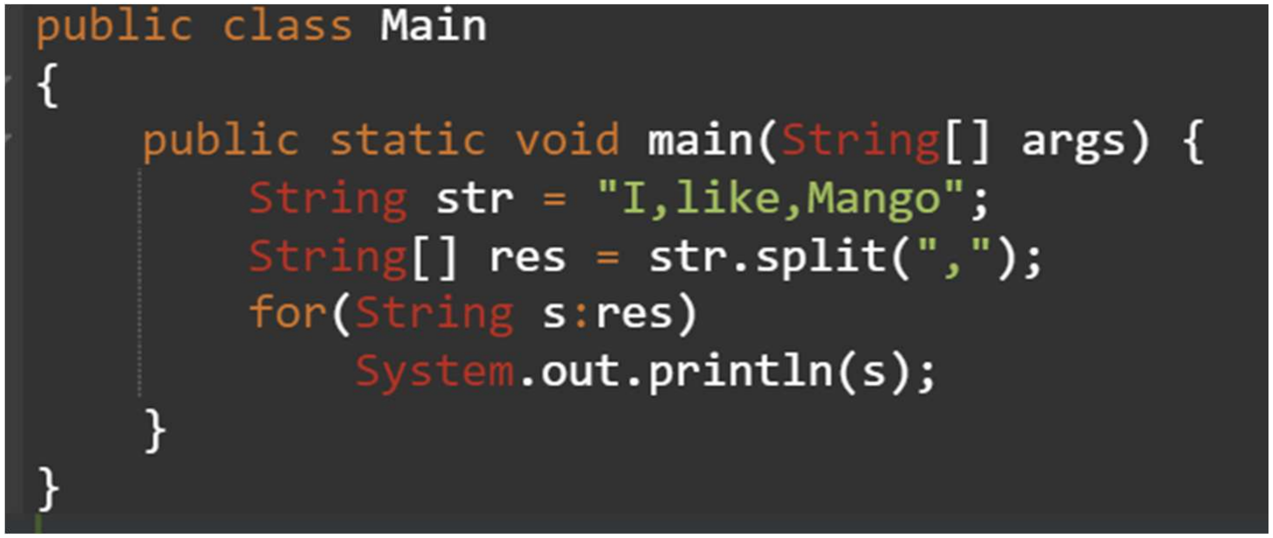
```
I
like
Mango
```

CRC-Training

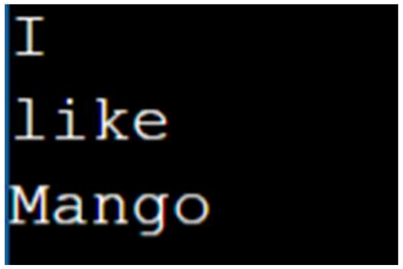
Inbuilt methods - split()

split() method splits this string against given regular expression and returns a char array.

splits the string based on comma



```
public class Main
{
    public static void main(String[] args) {
        String str = "I,like,Mango";
        String[] res = str.split(",");
        for(String s:res)
            System.out.println(s);
    }
}
```



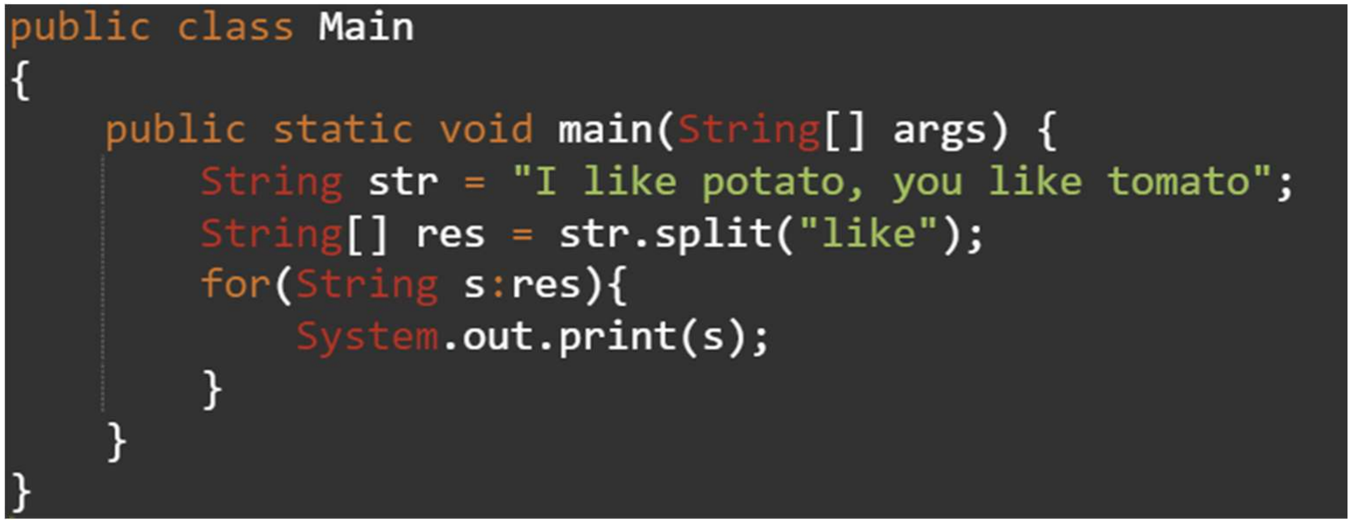
```
I
like
Mango
```

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

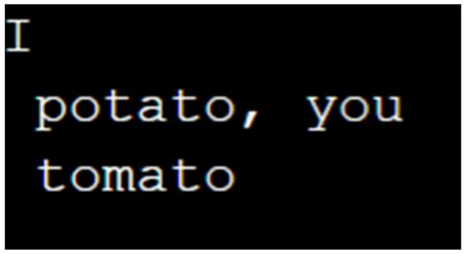
Inbuilt methods - split()

split() method splits this string against given regular expression and returns a char array.

splits the string based on sub string



```
public class Main
{
    public static void main(String[] args) {
        String str = "I like potato, you like tomato";
        String[] res = str.split("like");
        for(String s:res){
            System.out.print(s);
        }
    }
}
```



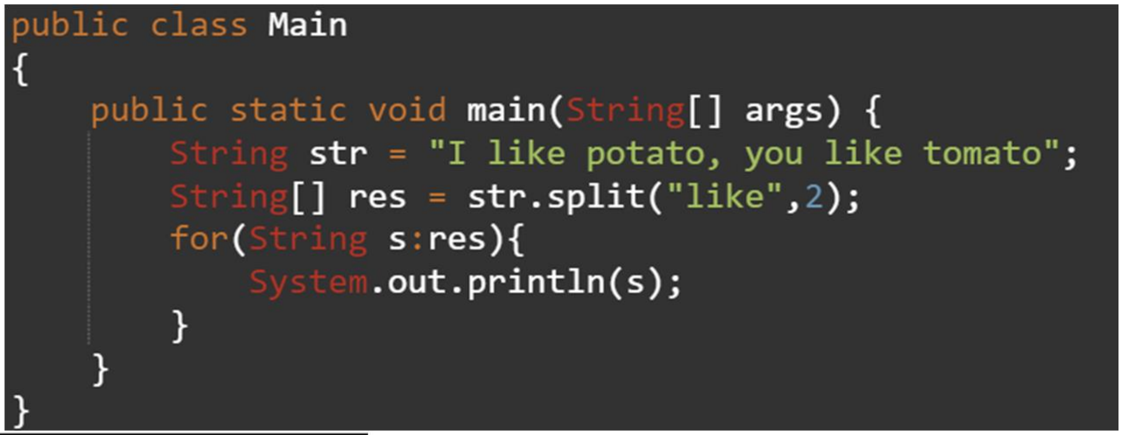
```
I
 potato, you
 tomato
```

ABESIT COLLEGE OF ENGINEERING
CRC-Training

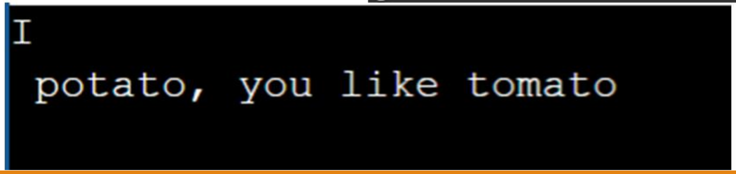
Inbuilt methods - split()

split(String regex, int limit) -

We can pass Limit also - limit for the number of strings in array. If it is zero, it will returns all the strings matching regex.



```java
public class Main
{
    public static void main(String[] args) {
        String str = "I like potato, you like tomato";
        String[] res = str.split("like",2);
        for(String s:res){
            System.out.println(s);
        }
    }
}
```



```
I
 potato, you like tomato
```

ABESIT COLLEGE OF ENGINEERING

CRC-Training

Inbuilt methods – indexOf()

indexOf() method returns the position of the first occurrence of the specified character or string in a specified string.

String str = "I am what I am";

0	1	2	3	4	5	6	7	8	9	10	11	12	13
I		**a**	**m**		**w**	**h**	**a**	**t**		**I**		**a**	**m**

str.indexOf('I') → 0

It returns the index position for the given char value

str.indexOf('I',2) → 10

// It returns the index position for the given char value and from index

ABESIT COLLEGE OF ENGINEERING

CRC-Training

Inbuilt methods – indexOf()

indexOf() method returns the position of the first occurrence of the specified character or string in a specified string.

String str = "I am what I am";

0	1	2	3	4	5	6	7	8	9	10	11	12	13
I		**a**	**m**		**w**	**h**	**a**	**t**		**I**		**a**	**m**

str.indexOf("am") → 2

It returns the index position for the given substring

str.indexOf("am",5) → 12

// It returns the index position for the given substring and from index

CRC-Training

Some more methods

Str.toLowerCase()	It returns a string in lowercase.
Str. toUpperCase()	**It returns a string in uppercase.**
Str.trim()	**It removes beginning and ending spaces of this string.**
String.valueOf(str)	**converts different types of values into string. int to string, long to string, boolean to string, character to string, float to string, double to string, char array to string**

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Some more methods

Str.compareTo(str1, str2)	compares the given strings in the order of occurrence in the dictionary and returns 0 , -ve value or +ve value
Str.equals(anotherString)	used to verify if both the strings are equal or not. Returns True or False
Str.join(joiner, str1, str2, str3,..)	is used to join a group of strings using the joiner between them. The joiner variable can be any character, string or a sequence of characters.
Str. startsWith()	checks whether the string begins with the specified string or not.

CRC-Training

Problems

1. Write a program to create a new string repeating every character twice of a given string

2. WAP to print no of words in a string.

3. WAP to count number of vowels in a string.

4. Write a Java program to get the character at the given index within the String.

5. Write a program to find total number of alphabets, digits or special character in a string.

6. Write a Java program to test if a given string contains the specified sequence of char values.

7. Write a Java program to compare two strings lexicographically. Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions.

8. Write a Java program to check a given string is palindrome or not.

9. Write a Java program to check whether two String objects contain the same data.