University of Science and Technology Chittagong

Department of Computer Science and Engineering

Lab Task 4

Object Oriented Programming (Java)

Programming Basics

Course Instructor: Debabrata Mallick

Submitted By: Reya Moni

Student ID: 0022520005101026

Roll No: 25070126

Semester: 2nd Semester 45th Batch

Submission Date:30/01/2026

Java Strings

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String greeting = "Hello";
        System.out.println(greeting);
    }
}
```

Code & Output

```
Hello
PS C:\Users\asus\OneDrive\Desktop\java project,r>
```

Strings are used for storing text.

A `String` variable contains a collection of characters surrounded by double quotes (`""`).

String Length

```java
public class AllStrings {
    public static void main(String[] args) {
        String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        System.out.println("The length of the txt string is: " +
    }
}
```

Code & Output

```
The length of the txt string is: 26
PS C:\Users\asus\OneDrive\Desktop\java project,r>
```

A String in Java is actually an object, which means it contains **methods** that can perform certain operations on strings.

For example, you can find the length of a string with the `length()` method:

More String Methods

```java
public class AllStrings {
    public static void main(String[] args) {
        String txt = "Have a Good Day ";
        System.out.println(txt.toUpperCase());
        System.out.println(txt.toLowerCase());
    }
}
```

Code & Output

```
HAVE A GOOD DAY
have a good day
```

There are many string methods available in Java.

For example:

- The `toUpperCase()` method converts a string to **upper case** letters.
- The `toLowerCase()` method converts a string to **lower case** letters.

Finding a Character in a String

AllStrings.java

```
AllStrings.java > AllStrings > main(String[])
1   public class AllStrings {
        Run | Debug
2       public static void main(String[] args) {
3       String txt = "Please locate where 'locate' Occurs!";
4       System.out.println(txt.indexOf(str: "locate"));
5       }
6   }
```

Code & Output

```
7
```

The `indexOf()` method returns the **index** (the position) of the first occurrence of a specified text in a string (including whitespace).

Comparing Strings

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String txt1 = "Hello";
        String txt2 = "Hello";

        String txt3 = "Greetings";
        String txt4 = "Great things";

        System.out.println(txt1.equals(txt2));
        System.out.println(txt3.equals(txt4));
    }
}
```

Code & Output

```
true
false
```

equals() compares the **content** of two strings.
So "Hello" equals "Hello" gives **true**, but "Greetings" and "Great things" are different, so it gives **false**.

Removing Whitespace

AllStrings.java

AllStrings.java > AllStrings > main(String[])

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
     String txt = "   Hello World   ";
     System.out.println("Before: [" + txt + "]");
     System.out.println("After:  [" + txt.trim() + "]");
    }
}
```

Code & Output

```
Before: [   Hello World   ]
After:  [Hello World]
```

The `trim()` method removes whitespace from the beginning and the end of a string.

String Concatenation

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String args[]) {
      String firstName = "John";
      String lastName = "Doe";
      System.out.println(firstName + " " + lastName);
    }
}
```

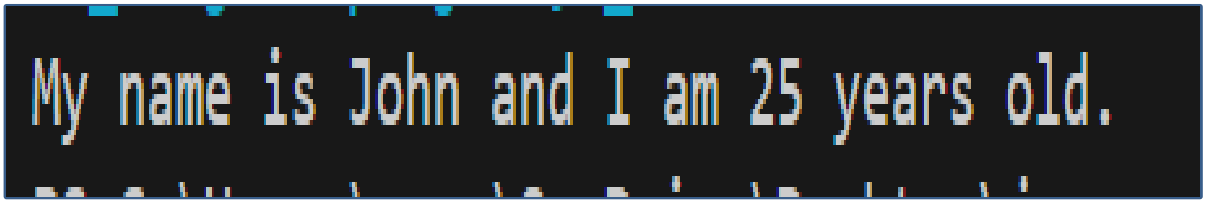
Code & Output

The + operator can be used between strings to combine them. This is called **concatenation**.

Java Numbers and Strings

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
      String name = "John";
      int age = 25;
      System.out.println("My name is " + name + " and I am " +
    }
}
```



Java uses the + operator for both addition and concatenation.

Numbers are added. Strings are concatenated.

Java Special Characters

, is to use the **backslash escape character**.

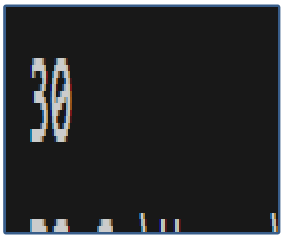
The backslash (\) escape character turns special characters into string characters.

The sequence \' inserts a single quote in a string.

The sequence \\ inserts a single backslash in a string.

AllStrings.java

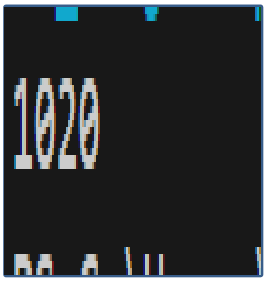
```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int z = x + y;
        System.out.println(z);
    }
}
```



Code & Output

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String x = "10";
        String y = "20";
        String z = x + y;
        System.out.println(z);
    }
}
```



AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String txt = "We are the so-called \"Vikings\" from the north.";
        System.out.println(txt);
    }
}
```

Code & Output

```
We are the so-called "Vikings" from the north.
```

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String txt = "It\'s alright.";
        System.out.println(txt);
    }
}
```

```
It's alright.
```

AllStrings.java

```java
public class AllStrings {
    Run | Debug
    public static void main(String[] args) {
        String txt = "The character \\ is called backslash.";
        System.out.println(txt);
    }
}
```

```
The character \ is called backslash.
```

END