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 Booleans

```java
public class MyBool {
    public static void main(String[] args) {
        boolean isJavaFun = true;
        boolean isFishTasty = false;
        System.out.println(isJavaFun);
        System.out.println(isFishTasty);
    }
}
```

```
true
false
```

Very often in programming, need a data type that can only have one of two values, like:

- YES / NO
- ON / OFF
- TRUE / FALSE

For this, Java has a `boolean` data type, which can store `true` or `false` values.

A boolean type is declared with the `boolean` keyword and can only take the values `true` or `false.`

Boolean Expressions

```java
public class MyBool {
    public static void main(String[] args) {
        int x = 10;
        int y = 9;
        System.out.println(x > y); // returns true,
    }
}
```

```
true
```

A boolean expression returns a boolean value: `true` or `false`.

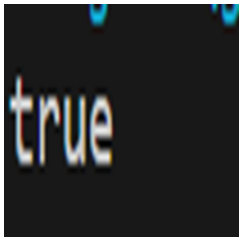
This is useful to build logic and make decisions in programs.

For example, you can use a comparison operator, such as the **greater than** (>) operator, to find out if an expression (or a variable) is true or false.

Real Life Example

MyBool.java

```java
public class MyBool {
    Run | Debug
    public static void main(String[] args) {
     int myAge = 25;
     int votingAge = 18;
     System.out.println(myAge >= votingAge);
    }
}
```



Let's think of a "real life example" where we need to find out if a person is old enough to vote.

In the example below, we use the >= comparison operator to find out if the age (25) is **greater than** OR **equal to** the voting age limit, which is set to 18.

END