

# LAB 2

---

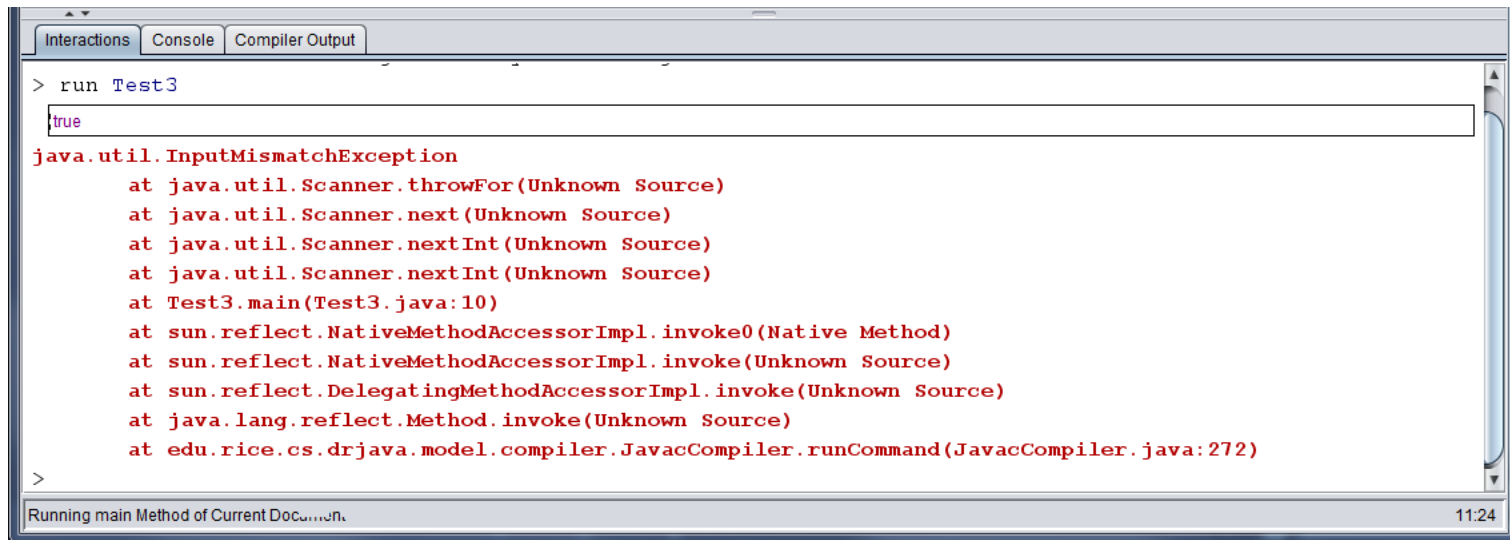
# Exercise 1: Interactive Programming

- Write a program using scanner to get the string input from the console and print the input number

```
public class MyHelloWorld{  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        int x = sc.nextInt();  
        System.out.println(x);  
    }  
}
```

# Exercise 1

- Let's give an input of type
  - String
  - Boolean



The screenshot shows an IDE console window with three tabs: "Interactions", "Console", and "Compiler Output". The "Console" tab is active. The prompt "> run Test3" is visible. Below it, the input "true" is shown in a text field. The console output displays a `java.util.InputMismatchException` stack trace. The stack trace starts with `java.util.InputMismatchException` and lists several frames, including `at java.util.Scanner.throwFor(Unknown Source)`, `at java.util.Scanner.next(Unknown Source)`, `at java.util.Scanner.nextInt(Unknown Source)`, `at java.util.Scanner.nextInt(Unknown Source)`, `at Test3.main(Test3.java:10)`, `at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)`, `at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)`, `at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)`, `at java.lang.reflect.Method.invoke(Unknown Source)`, and `at edu.rice.cs.drjava.model.compiler.JavacCompiler.runCommand(JavacCompiler.java:272)`. The status bar at the bottom indicates "Running main Method of Current Document." and the time "11:24".

```
> run Test3
true
java.util.InputMismatchException
    at java.util.Scanner.throwFor(Unknown Source)
    at java.util.Scanner.next(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at Test3.main(Test3.java:10)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at edu.rice.cs.drjava.model.compiler.JavacCompiler.runCommand(JavacCompiler.java:272)
>
```

Running main Method of Current Document. 11:24

# Exercise 1

- Add statements to get and print a Boolean input, Double input, and String input

```
public class Lab3{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        System.out.println(x);
        boolean w = sc.nextBoolean();
        System.out.println(w);
        double y = sc.nextDouble();
        System.out.println(y);
        String z = sc.next();
        System.out.println(z);
    }
}
```

# Exercise 1

- Modify a program to obtain integers from one line of input

```
public class Lab3{  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        int x = sc.nextInt();  
        System.out.println(x);  
        int w = sc.nextInt();  
        System.out.println(w);  
        int y = sc.nextInt();  
        System.out.println(y);  
        int z = sc.nextInt();  
        System.out.println(z);  
    }  
}
```

# Exercise 1

- Modify a program to obtain strings from one line of input

```
public class Lab3{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String x = sc.next();
        System.out.println(x);
        String w = sc.next();
        System.out.println(w);
        String y = sc.next();
        System.out.println(y);
        String z = sc.next();
        System.out.println(z);
    }
}
```

# Exercise 1

- Conclusion:
  - Scanner will wait for an input from the specified source
    - `System.in` -> console/keyboard
  - `next()` method gets a token of input, separated by space
    - `next()` get a string token
    - `nextInt()` get an integer token
    - `nextBoolean()` get a boolean token
    - `nextDouble()` get double token
    - `nextLine()` get a whole line of string input

# Exercise 2: String

- Declare new string

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("Hello");  
        String st2 = "Programming";  
        System.out.println(st1);  
        System.out.println(st2);  
    }  
}
```

- Two form can be used:
  - String st = new String("Hello");
  - String st = "Hello";



# Exercise 2

- Try the following methods on the string st2

```
st2.length();  
st2.charAt(3);  
st2.concat(st1);  
st2.indexOf("a");  
st2.indexOf("m", 7);  
st2.substring(3);  
st2.substring(3, 6);
```

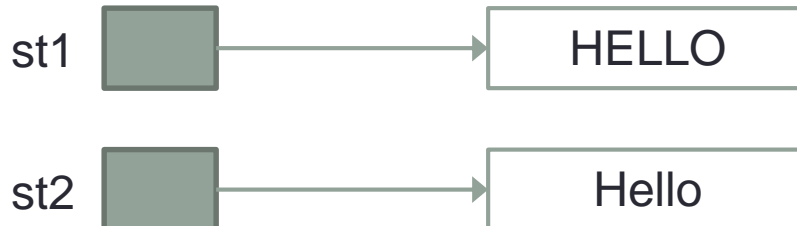
- What is the result of each statement?

# Exercise 3: String Comparison

- Declare two strings containing the same text but different case

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("HELLO");  
        String st2 = "Hello";  
        System.out.println(st1==st2);  
    }  
}
```

- Are they equal?

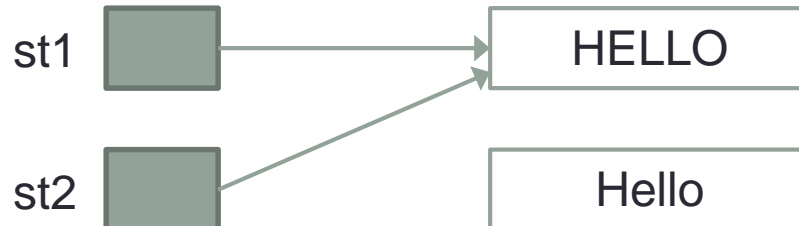


# Exercise 3

- Assign `st2 = st1`

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("HELLO");  
        String st2 = "Hello";  
        st2 = st1;  
        System.out.println(st1==st2);  
    }  
}
```

- Are they equal?



# Exercise 3

- CompareTo

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("HELLO");  
        String st2 = "Hello";  
        System.out.println(st1.compareTo(st2));  
        System.out.println(st1.compareToIgnoreCase(st2));  
    }  
}
```

- Conclusion:

- CompareTo is case sensitive
- CompareToIgnoreCase is case insensitive (ignore case)

# Exercise 3

- CompareTo will give
  - Negative result if st1 comes before st2
  - 0 if both string are the same
  - Positive result if st1 comes after st2
- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("ape");  
        String st2 = "zebra";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- -25 why?
  - a comes before z -> negative value
  - In Unicode, a = 97 and z = 122, then a - z = -25

# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("zebra");  
        String st2 = "ape";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- 25 why?
  - z comes after a -> positive value
  - In Unicode, a = 97 and z = 122, then  $z - a = 25$

# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("ape");  
        String st2 = "ape";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- 0 why?
  - $a = a \rightarrow 0$
  - In Unicode,  $a = 97$  then  $a - a = 0$

# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("ape");  
        String st2 = "apple";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- -11 why?
  - e vs. p
  - e comes before p -> negative value
  - In Unicode, e = 101 and p = 112 then e - p = -11



# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("Ape");  
        String st2 = "ape";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- -32 why?
  - In Unicode, A = 65 and a = 97, then A - a = -32

# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String(" ape");  
        String st2 = "Ape";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- -33 why?
  - In Unicode, space = 32 and A = 65, then space - A = -33

# Exercise 3

- Try the following, what is the result?

```
public class Lab3{  
    public static void main(String[] args){  
        String st1 = new String("cat");  
        String st2 = "catfish";  
        System.out.println(st1.compareTo(st2));  
    }  
}
```

- -4 why?
  - cat comes before catfish -> negative value
  - catfish has four letters longer than cat, then Java return the number of different characters which is -4

# Exercise 4: If-Else Statement

- Write the if statement

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 10;  
        if(x==10){  
            System.out.println("x is equal to 10");  
        }  
    }  
}
```

# Exercise 4

- Then add an else statement, to print the value of x

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 10;  
        if(x==10){  
            System.out.println("x is equal to 10");  
        }  
        else{  
            System.out.println("x is "+ x);  
        }  
    }  
}
```

# Exercise 4

- Write a conditional operator to express the previous if-else statement

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 10;  
        System.out.println(x==10? "x is equal to 10": "x is "+x);  
    }  
}
```

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 10;  
        if(x==10){  
            System.out.println("x is equal to 10");  
        }  
        else{  
            System.out.println("x is "+ x);  
        }  
    }  
}
```

# Exercise 4

- Multiple conditions

```
public class Lab3{
    public static void main(String[] args){
        int x = 10;
        if(x==10){
            System.out.println("x is equal to 10");
        }
        else if(x<10){
            System.out.println("x is less than 10");
        }
        else if(x>10 && x<50){
            System.out.println("x is greater than 10");
        }
        else{
            System.out.println("x is " + x);
        }
    }
}
```

# Exercise 5: AND/OR Operator

- AND/OR operator vs. Short-circuited AND/OR
  - AND (&), OR (|)
  - Short-circuited AND (&&), Short-circuited OR (||)

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 1;  
        boolean result1 = (x > 1) & (x++ < 10);  
        System.out.println(result1);  
        System.out.println(x);  
  
        x = 1;  
        boolean result2 = (x > 1) && (x++ < 10);  
        System.out.println(result2);  
        System.out.println(x);  
    }  
}
```



# Exercise 5

- Try the following expression

If x is 1, what is x after this expression?

`(1 > x) && (1 > x++)`

`(1 > x) & (1 > x++)`

`(1 > x) || (1 > x++)`

`(1 > x) | (1 > x++)`

How about

`(1 == x) || (10 > x++)`

`(1 == x) | (10 > x++)`

# Exercise 6: Switch Statement

- Write the Switch statement as follows and run:

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 2;  
        switch(x){  
            case 1:  
                System.out.println("one");  
            case 2:  
                System.out.println("two");  
            case 3:  
                System.out.println("three");  
            case 4:  
                System.out.println("four");  
            case 5:  
                System.out.println("five");  
        }  
    }  
}
```

- What does it print?

# Exercise 6

- Put Break command at the first two cases

```
public class Lab3{
    public static void main(String[] args){
        int x = 2;
        switch(x){
            case 1:
                System.out.println("one"); break;
            case 2:
                System.out.println("two");break;
            case 3:
                System.out.println("three");
            case 4:
                System.out.println("four");
            case 5:
                System.out.println("five");
        }
    }
}
```

- What does it print?
- What if  $x = 3$ ?

# Exercise 6

- What if  $x = 7$ ?

```
public class Lab3{  
    public static void main(String[] args){  
        int x = 7;  
        switch(x){  
            case 1:  
                System.out.println("one"); break;  
            case 2:  
                System.out.println("two"); break;  
            case 3:  
                System.out.println("three"); break;  
            case 4:  
                System.out.println("four"); break;  
            case 5:  
                System.out.println("five"); break;  
        }  
    }  
}
```

- Nothing printed because no case matched

# Exercise 6

- Put a default case

```
int x = 7;
switch(x){
case 1:
    System.out.println("one"); break;
case 2:
    System.out.println("two"); break;
case 3:
    System.out.println("three"); break;
case 4:
    System.out.println("four"); break;
case 5:
    System.out.println("five"); break;
default:
    System.out.println("no matched"); break;
```

- What does it print?

# Exercise 6

- Use string

```
String x = "four";
switch(x) {
case "one":
    System.out.println(1); break;
case "two":
    System.out.println(2); break;
case "three":
    System.out.println(3); break;
case "four":
    System.out.println(4); break;
case "five":
    System.out.println(5); break;
default:
    System.out.println("no matched"); break;
```

- What does it print?

# DIY

- You can use the following source code as for your starting point

```
public class ClassName {  
    public static void main(String args[]){  
        Scanner sc = new Scanner(System.in);  
        int cases = sc.nextInt(); //get number of test cases  
        int count = 0;  
        while(count<cases){ //do your code repeatedly until count = 50  
            //Your code is Here  
  
            count++;  
        }  
    }  
}
```

# DIY

- User a scanner get input regarding your name, surname, age and study program. Then print those information as:
  - My name is ...(name)..(surname). .
  - I'm ...(age)...years old.
  - I'm studying ...(studyProgram)... .
- Get input of a sentence from scanner as String. Then:
  - Transform the sentence to lower cases and print
  - Transform the sentence to upper cases and print
  - Print length of the sentence
  - Print the letter at the index of 2
  - Print sub-string from the letter at the index of 2



# DIY

- Get two Strings from Scanner, then compare them
  - If the strings (words) are equal (ignore case), print “equal”
  - Else, print “not equal”
- Day in a week, use if-else to print day in a week
  - 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday, and 7 = Sunday.
- Day in a week, use switch function to print day in a week