Introduction to computer programming A
LAB3

Upload channel will open during class time, please upload a picture of you taking a live lab class to blackboard to represent your attendance.

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn

LAB OBJECTIVES

1. Practice storing values with primitive types
2. Learn how to obtain user input from command line arguments or by the Scanner class.
3. Learn how to use the if and if...else selection statements to choose between alternative actions.

1 Primitive types

Data type and its range

Table:List of Java's primitive data types

Type	Size in Bytes	Range
byte	1 byte	-2^7 to 2^7 -1 (-128 to 127)
short	2 bytes	-2^{15} to 2^{15} -1 (-32768 to 32767)
int	4 bytes	-2^{31} to 2^{31} -1 (-2147483648 to 2147483647)
long	8 bytes	-2^{63} to 2^{63} -1(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
float	4 bytes	approximately ±3.40282347E+38F (6-7 significant decimal digits) Java implements IEEE 754 standard
double	8 bytes	approximately ±1.79769313486231570E+308 (15 significant decimal digits)
char	2 byte	0 to 65,536 (unsigned)
boolean	not precisely defined*	true or false

*boolean represents one bit of information, but its "size" isn't something that's precisely defined

Data Operations(Basic Arithmetic Operators)

Name	*Meaning*	*Example*	*Result*
+	Addition	34 + 1	35
-	Subtraction	34.0 − 0.1	33.9
*	Multiplication	300*30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

```java
public class TestArithmeticOperators {
    public static void main(String[] args) {
        //Variables Definition and Initialization
          int number1 = 12, number2 = 4;
          //Addition Operation
          int sum = number1 + number2;
          System.out.println("Sum is: " + sum);
          //Subtraction Operation
          int dif = number1 - number2;
          System.out.println("Difference is : " + dif);
          //Multiplication Operation
          int mul = number1 * number2;
          System.out.println("Multiplied value is : " + mul);
          //Division Operation
          int div = number1 / number2;
          System.out.println("Quotient is : " + div);
          //Modulus Operation
          int rem = number1 % number2;
          System.out.println("Remainder is : " + rem);
    }
}
```

```
Sum is: 16
Difference is : 8
Multiplied value is : 48
Quotient is : 3
Remainder is : 0
```

Data Operations(Assignment Operators)

The Java Assignment Operators are used when you want to assign a value to the expression. The assignment operator denoted by the single equal sign =.

Syntax:

```
variable = expression;
```

Example:

```
int a = 6;
float b = 6.8F;
```

Data type Conversions

◆Widening or Automatic Type Conversion

- The two data types are compatible.(char and boolean are not compatible with each other.)
- When we assign value of a smaller data type to a bigger data type.

Byte –> Short –> Int –> Long – > Float –> Double

Example:

```
class Test
{
    public static void main(String[] args)
    {
        int i = 100;

        //automatic type conversion
        long l = i;

        //automatic type conversion
        float f = l;
        System.out.println("Int value "+i);
        System.out.println("Long value "+l);
        System.out.println("Float value "+f);
    }
}
```

```
Int value 100
Long value 100
Float value 100.0
```

Data type Conversions

◆Narrowing or Explicit Conversion

- This is useful for incompatible data types where automatic conversion cannot be done.
- Assign a value of larger data type to a smaller data type

Double –> Float –> Long –> Int –> Short –> Byte

Example:

```
//Java program to illustrate incompatible data
// type for explicit type conversion
public class Test
{
  public static void main(String[] argv)
  {
    char ch = 'c';
    int num = 88;
    ch = num;
  }
}
```

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
        Type mismatch: cannot convert from int to char
```

Example:

```
//Java program to illustrate explicit type conversion
class Test
{
    public static void main(String[] args)
    {
        double d = 100.04;

        //explicit type casting
        long l = (long)d;

        //explicit type casting
        int i = (int)l;
        System.out.println("Double value "+d);

        //fractional part lost
        System.out.println("Long value "+l);

        //fractional part lost
        System.out.println("Int value "+i);
    }
}
```

Output:

```
Double value 100.04
Long value 100
Int value 100
```

2 Input

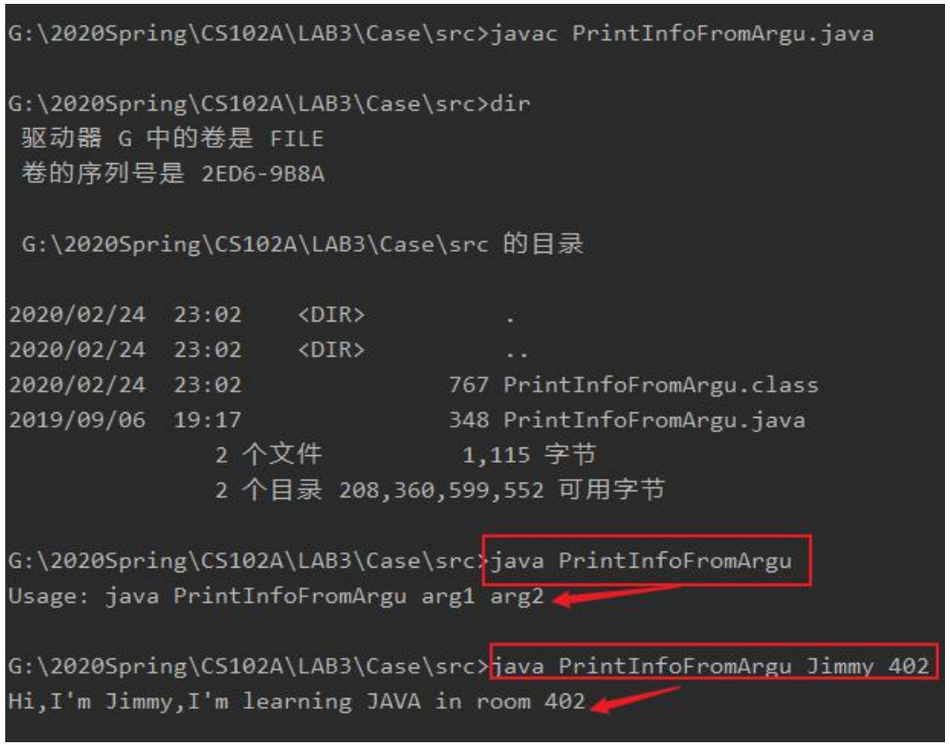
2.1 Running programs with arguments

2.2 Using Scanner to get input

2.3 Conclusion

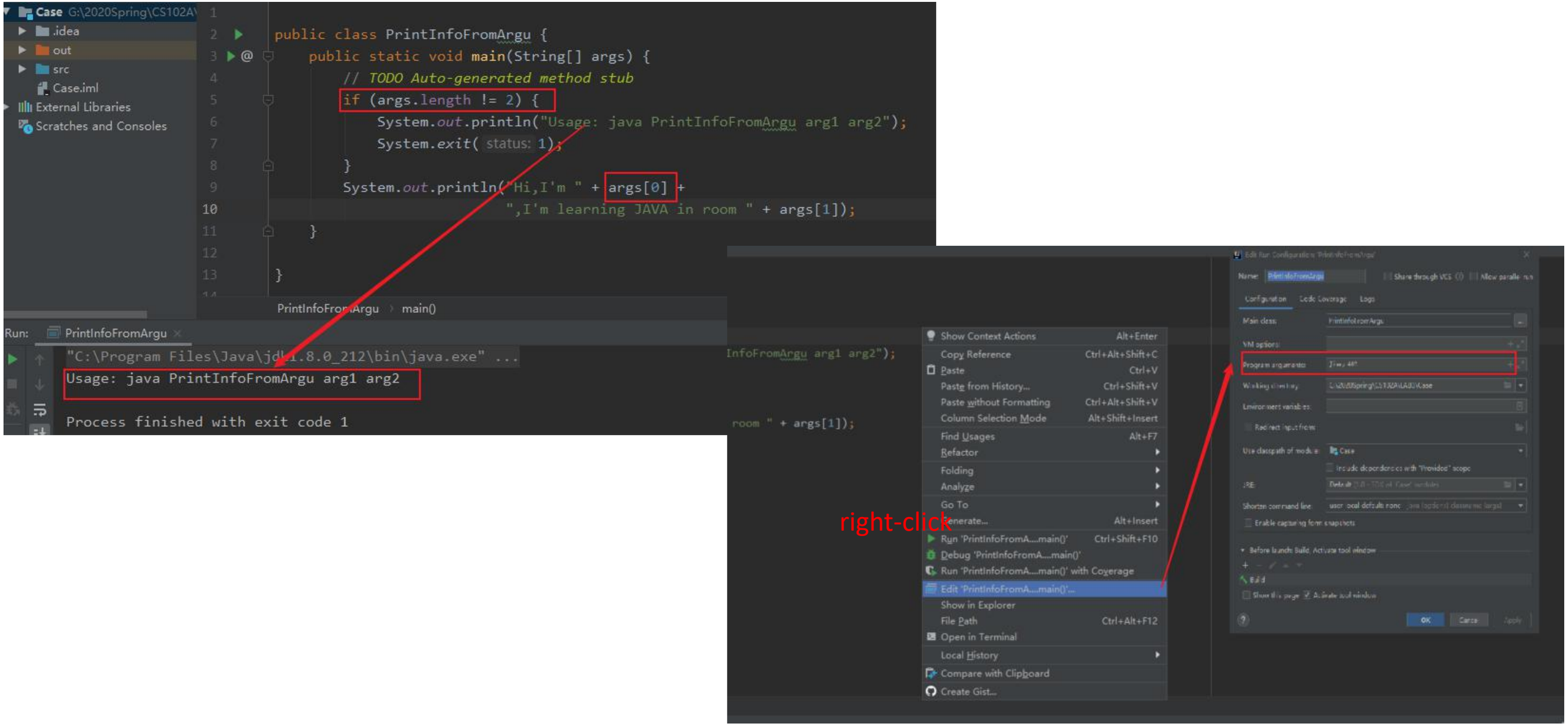
2.1 Runing programs with arguments

- **In command line**



2.1 Runing programs with arguments

- **In IDEA**



2.2 Using Scanner to get input

In you code

- 1. **import java.util.Scanner**
- 2. **new an Scanner object**
- 3. invoke the **method** of the Scanner object to get input data
  - 3-1: next() to get a string data
  - 3-2: nextInt() to get an integer data
  - 3-3: nextDouble() to get a double data
- 4. if all the input process is end, invoke the "**close()**" method is suggested to make your program safe

```java
import java.util.Scanner;
public class Demo_scanner{
    public static void main(String [] args){
        Scanner input=new Scanner(System.in);

        System.out.print("please input name: ");
        String name = input.next();
        System.out.print("please input age: ");
        int age = input.nextInt();
        System.out.print("please input level: ");
        char level = input.next().charAt(0);
        System.out.print("please input grade: ");
        double grade = input.nextDouble();

        System.out.printf("My name is %s.\nI am %d years old.\n"
        +"I got %c in Java last semester.\t My score is %.1f\n",name,age,level,grade);
    }
}
```

```
c:\vivian\Java_2018_spring\test>java Demo_scanner
please input name: John
please input age: 18
please input level: A+
please input grade: 96.5
My name is John.
I am 18 years old.
I got A in Java last semester.    My score is 96.5
```

4.3 Conclusion

While using argument of running

1) The arguments is stored in String [] args
2) You may need to change its type as your desired

```
public class Demo{
    public static void main(String [] args){
        int age= Integer.parseInt(args[0]);
        double grade= Double.parseDouble(args[1]);
        char level = args[2].charAt(0);
        String name = args[3];
```

While using Scanner object

1) Import scanner
2) New a object
3) Invoke the method to get the right type of data
4) Invoke close() is sugguested

```
import java.util.Scanner;
public class Demo_scanner{
    public static void main(String [] args){
        Scanner input=new Scanner(System.in);

        System.out.print("please input name: ");
        String name = input.next();
        System.out.print("please input age: ");
        int age = input.nextInt();
        System.out.print("please input level: ");
        char level = input.next().charAt(0);
        System.out.print("please input grade: ");
        double grade = input.nextDouble();
```

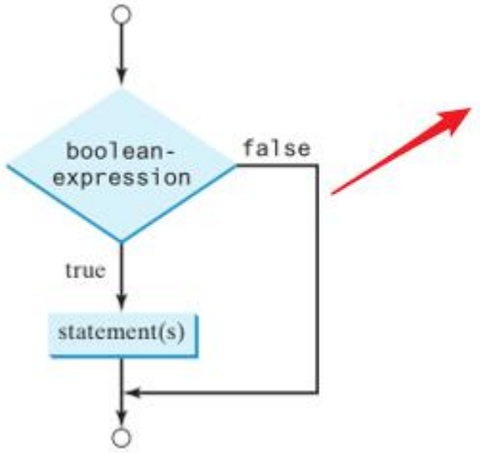
3 Selections

3.1 if Statements

3.2 if-else Statements

3.3 else-if Statements

3.1 if Statements(if)




```
if (boolean-expression) {
  statement(s);
}
```

Example:

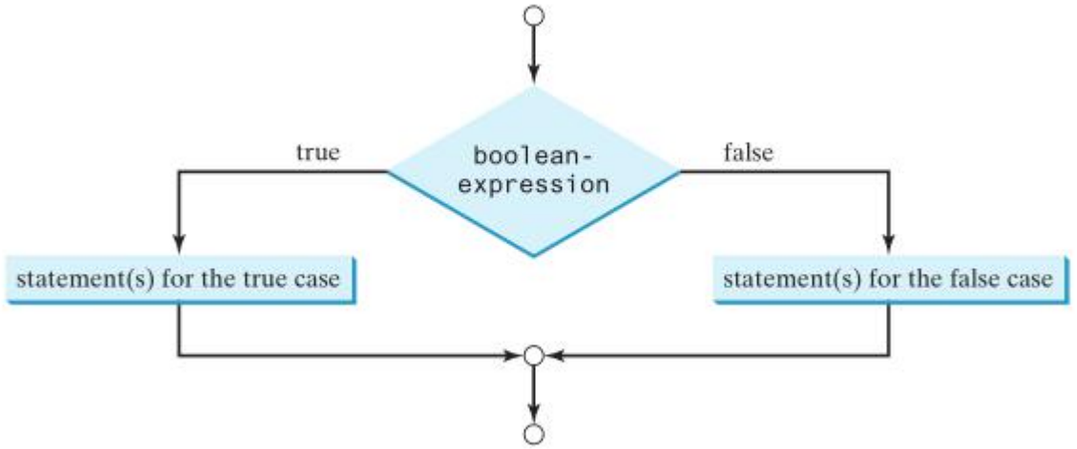
```
import java.util.Iterator;
import java.util.Scanner;

public class SimpleIfDemo {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        // enter input
        int number = input.nextInt();
        // check 5
        if (number % 5 == 0)
            System.out.println("HiFive");
        // check even
        if (number % 2 == 0)
            System.out.println("HiEven");

    }


}
```

Console

<terminated> SimpleIfDemo [Java Application] C

```
Enter an integer: 30
HiFive
HiEven
```

3.2 if Statements(if-else)




```
if (boolean-expression) {
  statement(s)-for-the-true-case;
}
else {
  statement(s)-for-the-false-case;
}
```

Example:

```
import java.util.Scanner;

public class SimpleIfelseDemo {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        // enter input
        int number = input.nextInt();
        if (number % 2 == 0)
            System.out.println(number + " is even.");
            else
            System.out.println(number + " is odd.");
    }

}
```

Console

<terminated> SimpleIfelseDemo [Java Appli

```
Enter an integer: 5
5 is odd.
```

Console

<terminated> SimpleIfelseDemo [Java Applicati

```
Enter an integer: 6
6 is even.
```

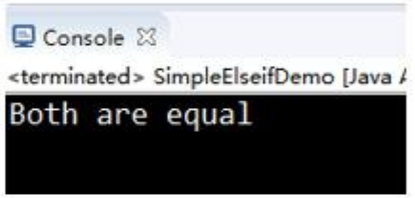
3.3 if Statements(else-if and Nested if)

```
if(boolean-expression)
{
    //execute your code
}
else if(boolean-expression n)
{
    //execute your code
}
else
{
    //execute your code
}
```

Example:

```
public class SimpleElseifDemo {

    public static void main(String[] args) {
        int a = 30, b = 30;
        if (b > a) {
            System.out.println("b is greater");
        }
        else if(a > b){
            System.out.println("a is greater");
        }
        else {
            System.out.println("Both are equal");
        }
    }
}
```

else-if



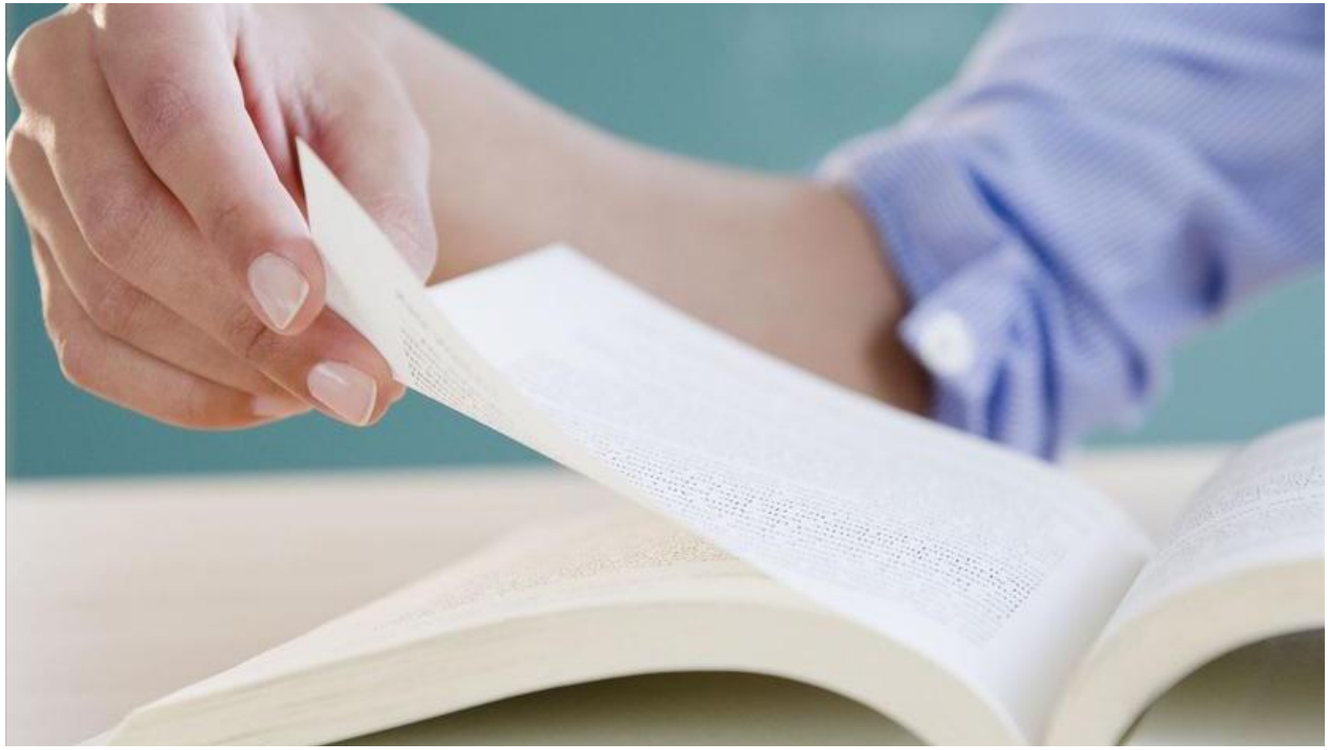

```
if(boolean-expression)
{
    if(boolean-expression)
     {
        //execute your code
     }
}
else
{
    //execute your code
}
```

Example:

```
public class NestedIfDemo {

    public static void main(String[] args) {
        int i = 30, k = 30, j = 30;
        if (i > k) {
            if (j > k)
                System.out.println("i and j are greater than k");
        } else
            System.out.println("i is less than or equal to k");
    }
}
```

Nested if

4 Exercises



Complete the exercises in the **2020S-Java-A-Lab-3.pdf** and submit to the blackboard as required.

Naming rules of Java

- **Capitalize the first letter** of each word in a **class name**. for example, the class names **ComputeArea** and **System**
- **Lowercase** the **first** letter of the first word and **capitalize** the first letter of subsequent words of **the variable name**, such as string, stringBuilder, etc.
- **Capitalize every** letter in a **constant**, and use **underscores** between words. for example, the constants PI and MAX_VALUE .

It is important to follow the naming conventions to make your programs easy to read.

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn