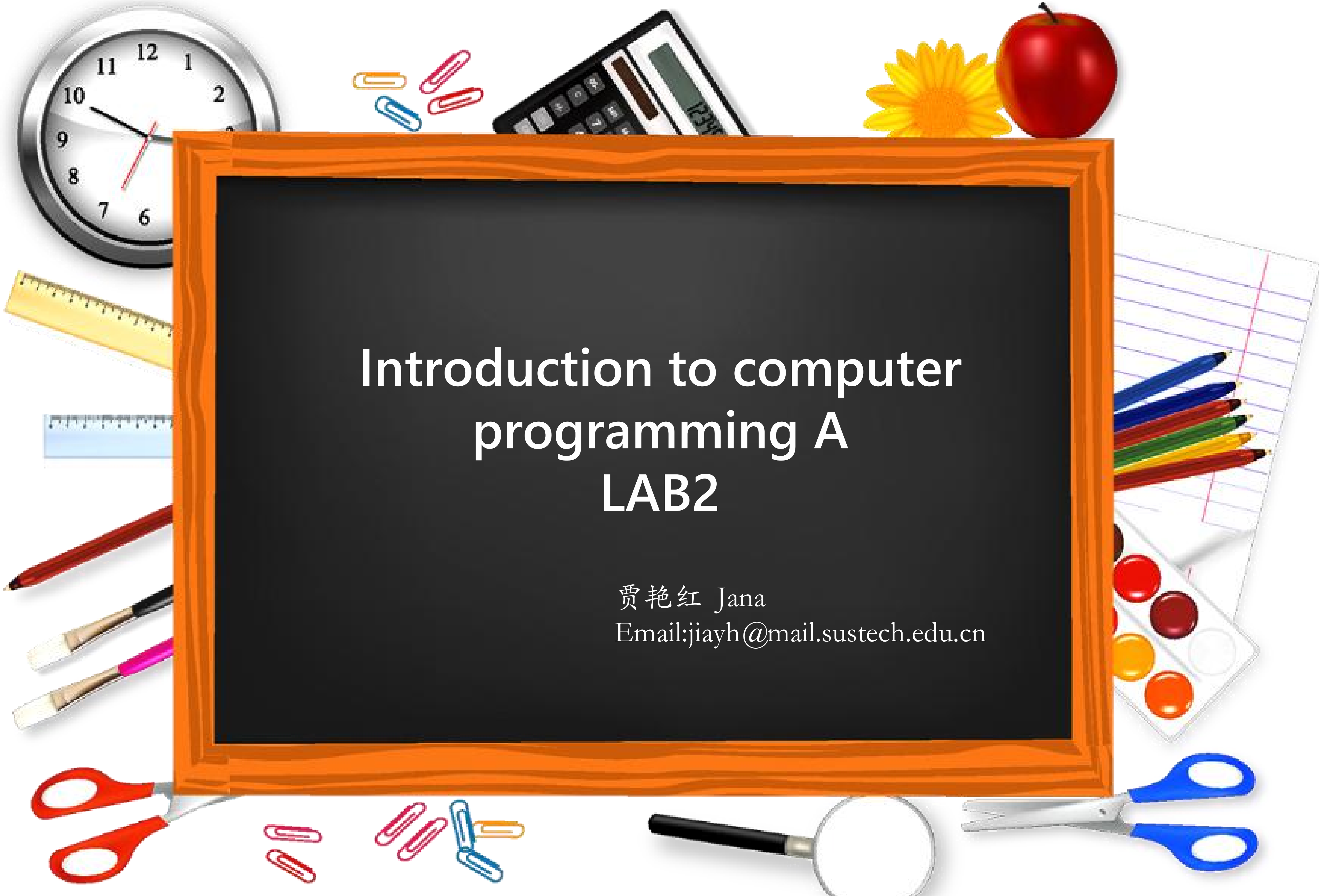


Introduction to computer programming A
LAB2

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

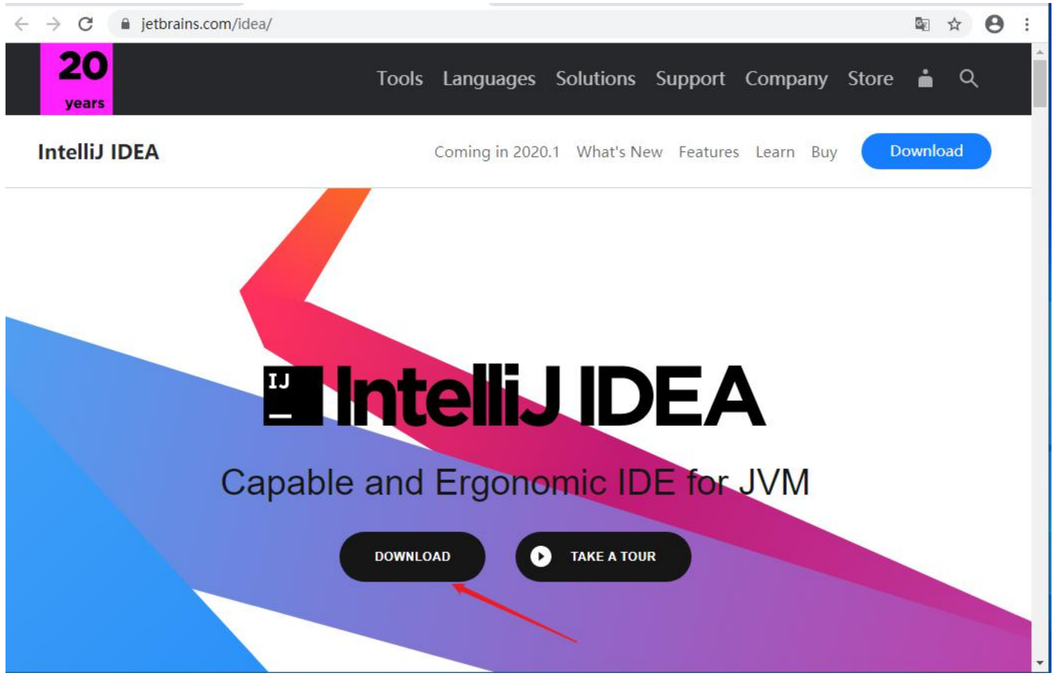
LAB OBJECTIVES

1. Learn how to use an Integrated Development Environment (IDE) in writing JAVA programs
2. Practice using input and output statements.
3. Practice storing values with primitive types

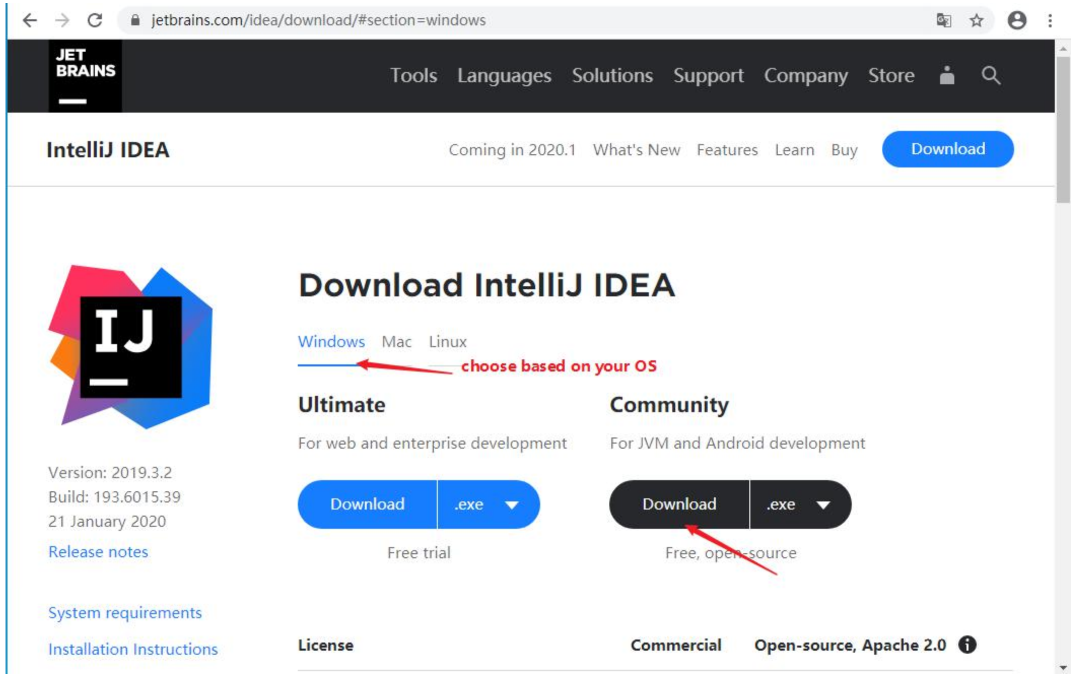
1 DownLoading and Installing IDEA

DownLoading IDEA

Download IDEA Installer from:https://www.jetbrains.com/idea/



Capable and Ergonomic IDE for JVM



Download IntelliJ IDEA

Ultimate

For web and enterprise development

Free trial

Community

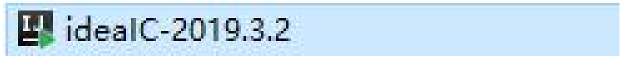
For JVM and Android development

Free, open-source

Version: 2019.3.2
Build: 193.6015.39
21 January 2020

License Commercial Open-source, Apache 2.0

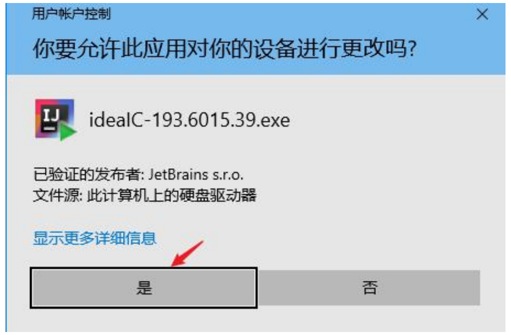
The idea installation package downloaded looks like this:



ideaIC-2019.3.2

Installing IDEA

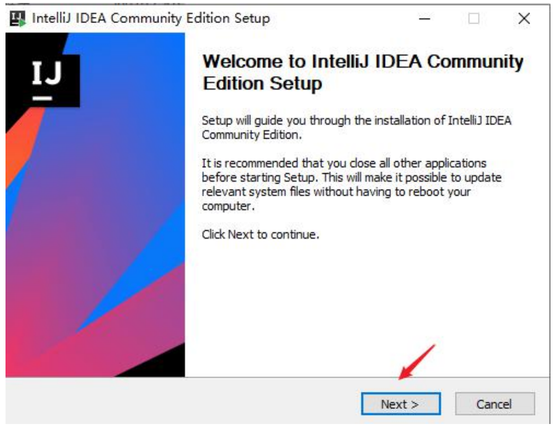
- Once the download is complete, locate the ***ideaIC-2019.3.2.exe*** file and **double-click to run the installer**.


用户帐户控制

你要允许此应用对你的设备进行更改吗?

ideaIC-193.6015.39.exe

已验证的发布者: JetBrains s.r.o.
文件源: 此计算机上的硬盘驱动器


IntelliJ IDEA Community Edition Setup

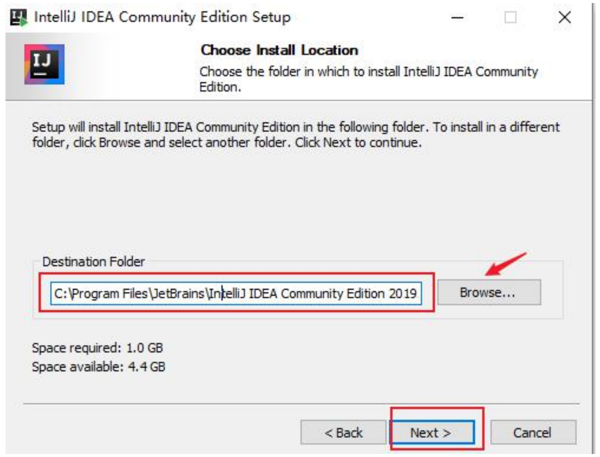
Welcome to IntelliJ IDEA Community Edition Setup

Setup will guide you through the installation of IntelliJ IDEA Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

- Click ***Next*** and on the following screen optionally change the installation location by clicking on the ***Change...*** button. In this example the default install location was kept.


IntelliJ IDEA Community Edition Setup

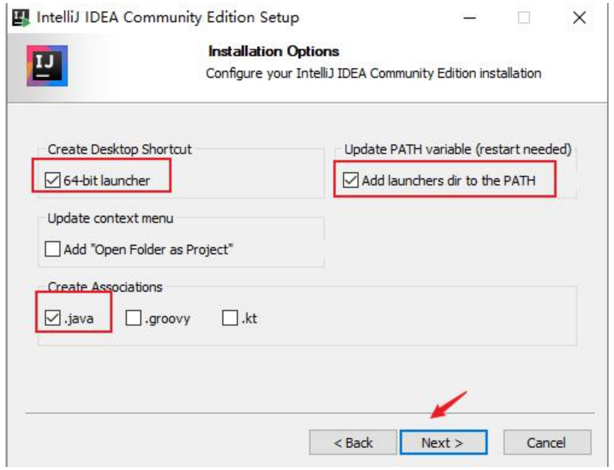
Choose Install Location

Choose the folder in which to install IntelliJ IDEA Community Edition.

Setup will install IntelliJ IDEA Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019

Space required: 1.0 GB
Space available: 4.4 GB


IntelliJ IDEA Community Edition Setup

Installation Options

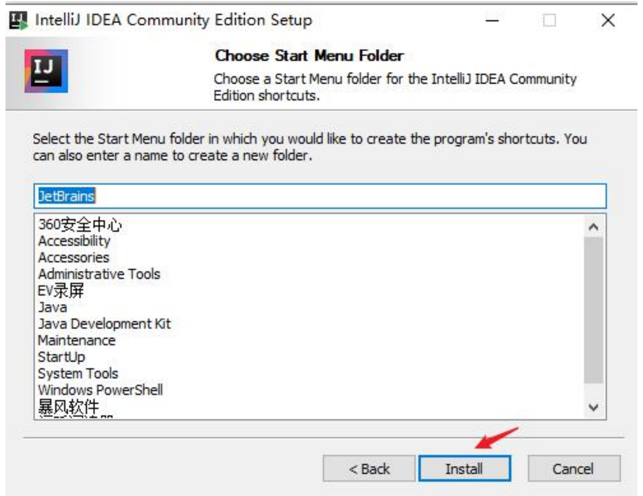
Configure your IntelliJ IDEA Community Edition installation

Create Desktop Shortcut
☑ 64-bit launcher

Update PATH variable (restart needed)
☑ Add launchers dir to the PATH

Update context menu
□ Add "Open Folder as Project"

Create Associations
☑ .java □ .groovy □ .kt


IntelliJ IDEA Community Edition Setup

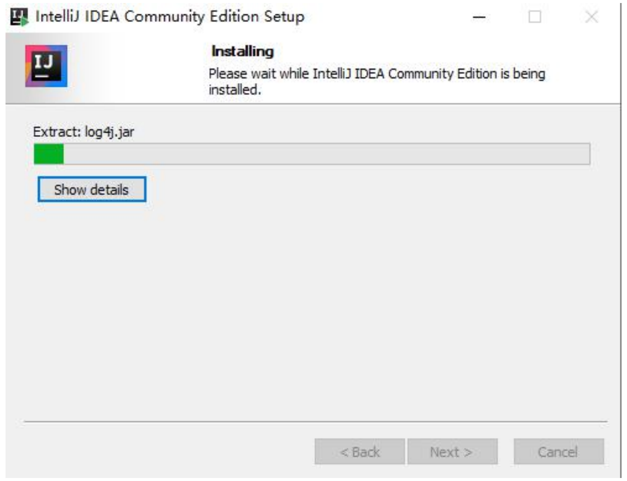
Choose Start Menu Folder

Choose a Start Menu folder for the IntelliJ IDEA Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

360安全中心
Accessibility
Accessories
Administrative Tools
EV录屏
Java
Java Development Kit
Maintenance
StartUp
System Tools
Windows PowerShell
暴风软件


IntelliJ IDEA Community Edition Setup

Installing

Please wait while IntelliJ IDEA Community Edition is being installed.

Extract: log4j.jar

Create a new Java project

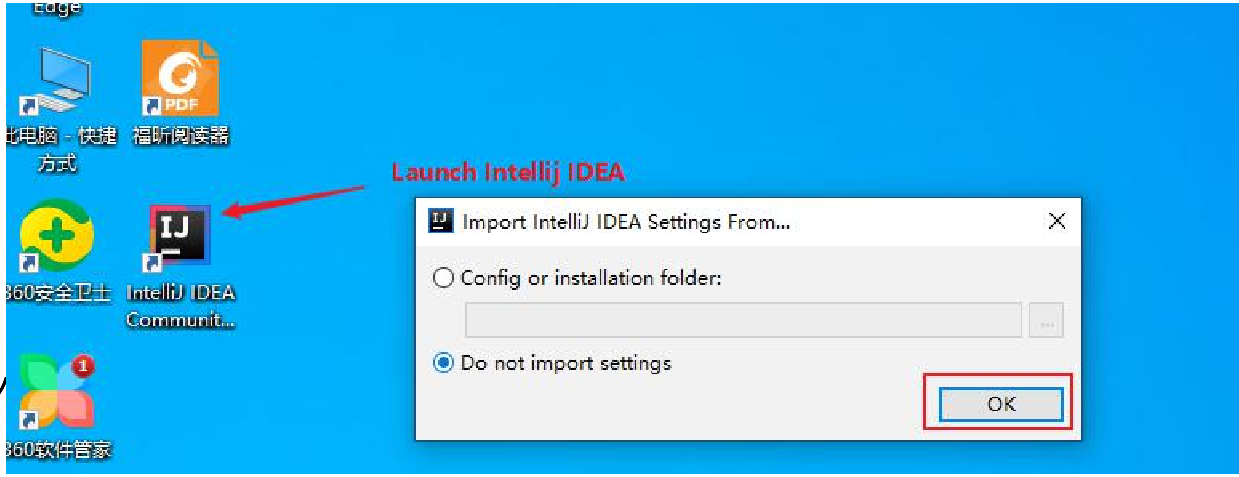
1. Launch IntelliJ IDEA.

If the Welcome screen opens, click Create New Project.Otherwise, from the main menu, select ***File | New | Project.***

2.In the **New Project** wizard, select Java from the list on the left.

3.From the **Project SDK** list, select the JDK that you want to use in your project.

*If the list is empty, click New and specify the path to the Java home directory (for example,**C:\Program Files\Java\jdk1.8.0_212**).*

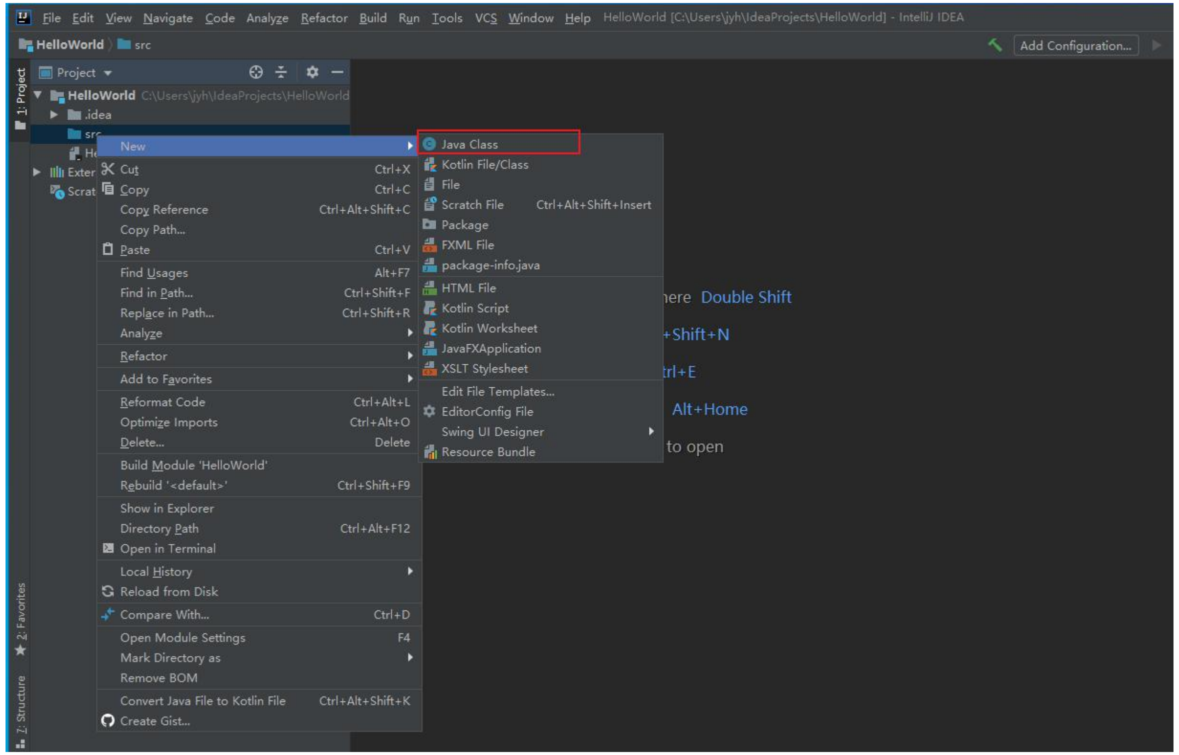


Launch Intellij IDEA

Import IntelliJ IDEA Settings From...

○ Config or installation folder:

● Do not import settings

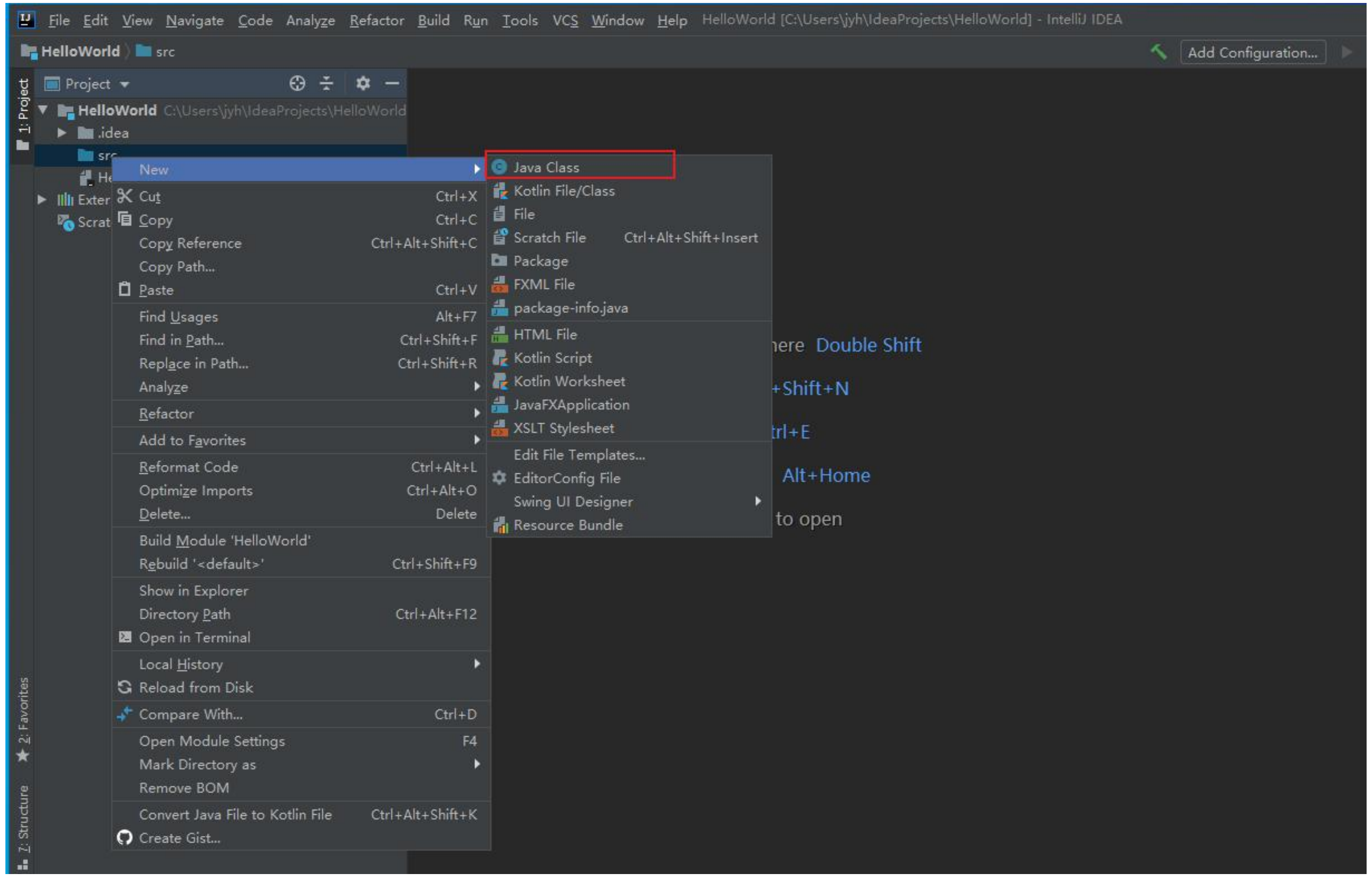


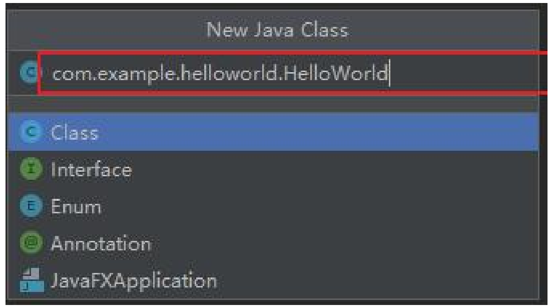
Create a package and a class

1. In the **Project** tool window, select the **src** folder, right-click on it, and select **Java Class**.

2. In the Name field, type ***com.example.helloworld.***HelloWorld and **click** OK.

*IntelliJ IDEA creates the **com.example.helloworld** package and the **HelloWorld** class.*





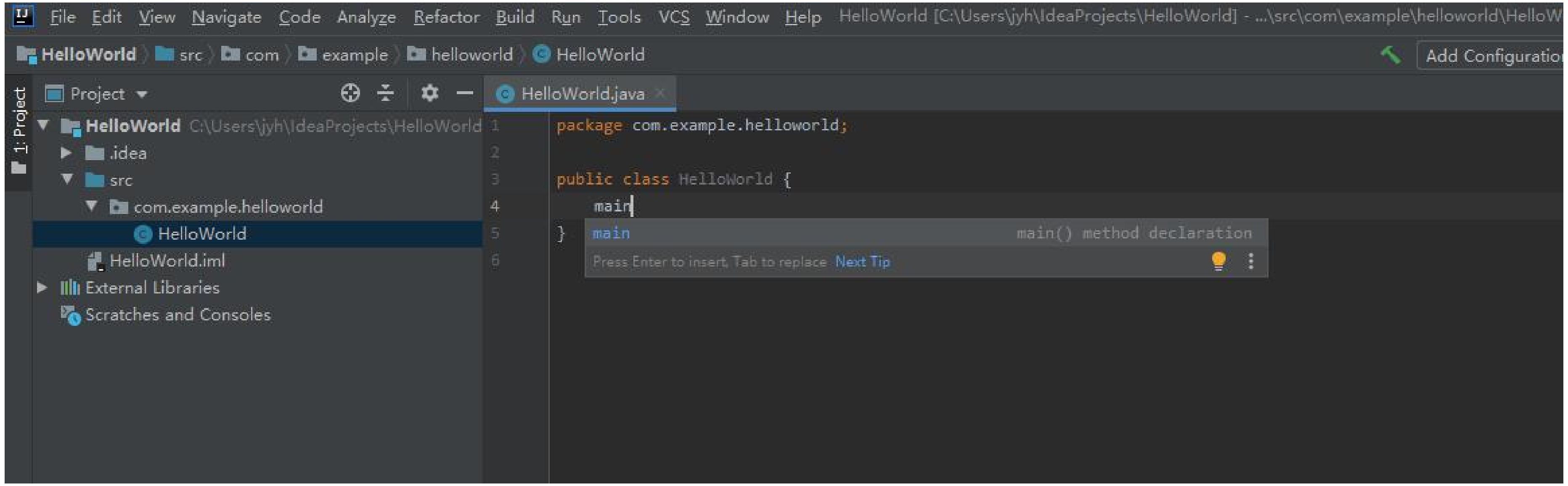
com.example.helloworld.HelloWorld

Note:
In Java, there's a **naming convention** that you should follow when you name packages and classes.

https://www.oracle.com/technetwork/java/codeconventions-135099.html

Write the code

1.Type **main** and select the template that inserts the **main**() method declaration.

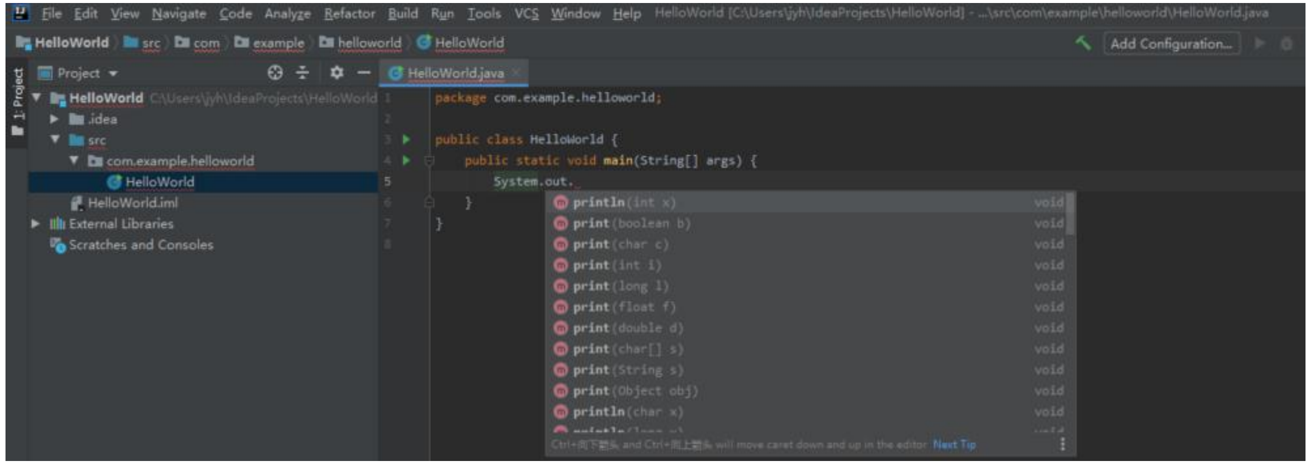

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HelloWorld [C:\Users\jyh\IdeaProjects\HelloWorld] - ...\src\com\example\helloworld\HelloW
HelloWorld src com example helloworld HelloWorld
Add Configuratio
Project
HelloWorld.java
1: Project
HelloWorld C:\Users\jyh\IdeaProjects\HelloWorld
.idea
src
com.example.helloworld
HelloWorld
HelloWorld.iml
External Libraries
Scratches and Consoles

```
package com.example.helloworld;

public class HelloWorld {
    main
}
```

main   main() method declaration

Press Enter to insert, Tab to replace Next Tip

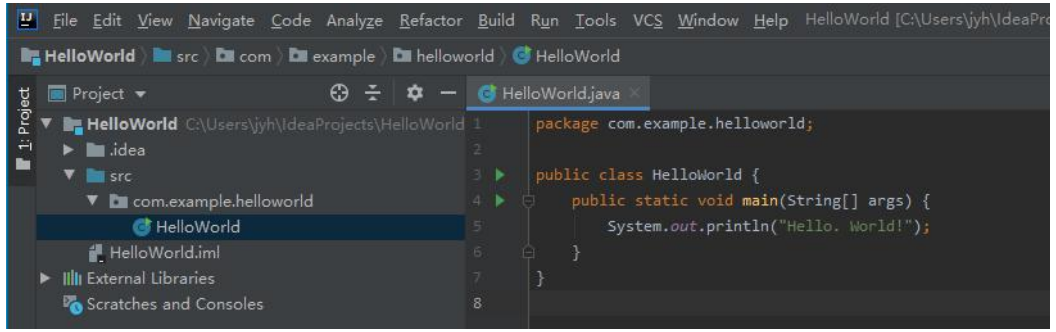
2.Call the **println**() method using code completion


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HelloWorld [C:\Users\jyh\IdeaProjects\HelloWorld] - ...\src\com\example\helloworld\HelloWorld.java
HelloWorld src com example helloworld HelloWorld
Add Configuration...
Project
HelloWorld.java
1: Project
HelloWorld C:\Users\jyh\IdeaProjects\HelloWorld
.idea
src
com.example.helloworld
HelloWorld
HelloWorld.iml
External Libraries
Scratches and Consoles

```
package com.example.helloworld;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.
    }
}
```

```
println(int x)        void
print(boolean b)      void
print(char c)         void
print(int i)          void
print(long l)         void
print(float f)        void
print(double d)       void
print(char[] s)       void
print(String s)       void
print(Object obj)     void
println(char x)       void
```

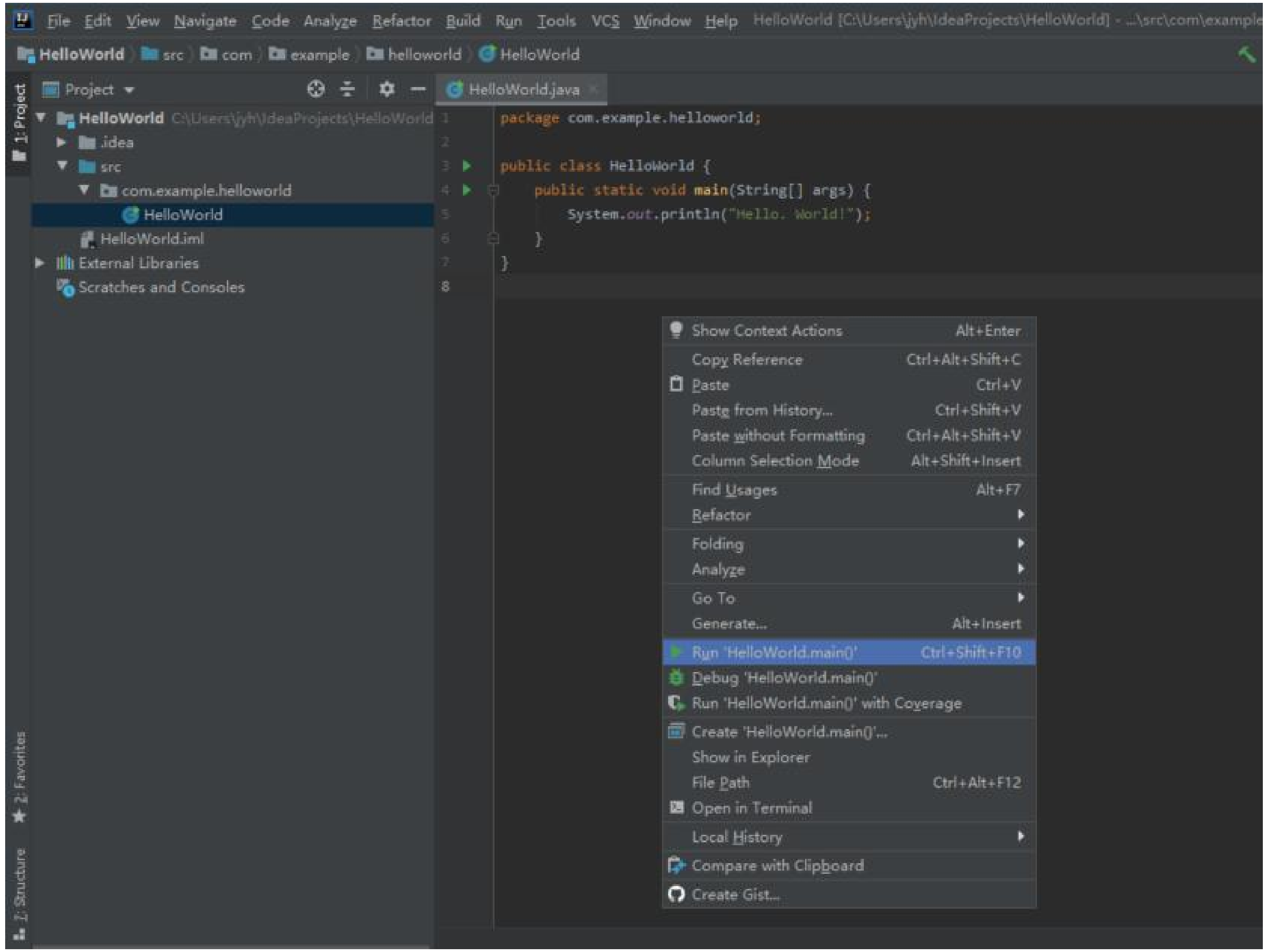
Ctrl+向下箭头 and Ctrl+向上箭头 will move caret down and up in the editor Next Tip


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HelloWorld [C:\Users\jyh\IdeaPro
HelloWorld src com example helloworld HelloWorld
Project
HelloWorld.java
1: Project
HelloWorld C:\Users\jyh\IdeaProjects\HelloWorld
.idea
src
com.example.helloworld
HelloWorld
HelloWorld.iml
External Libraries
Scratches and Consoles

```
package com.example.helloworld;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello. World!");
    }
}
```

Build and run the application

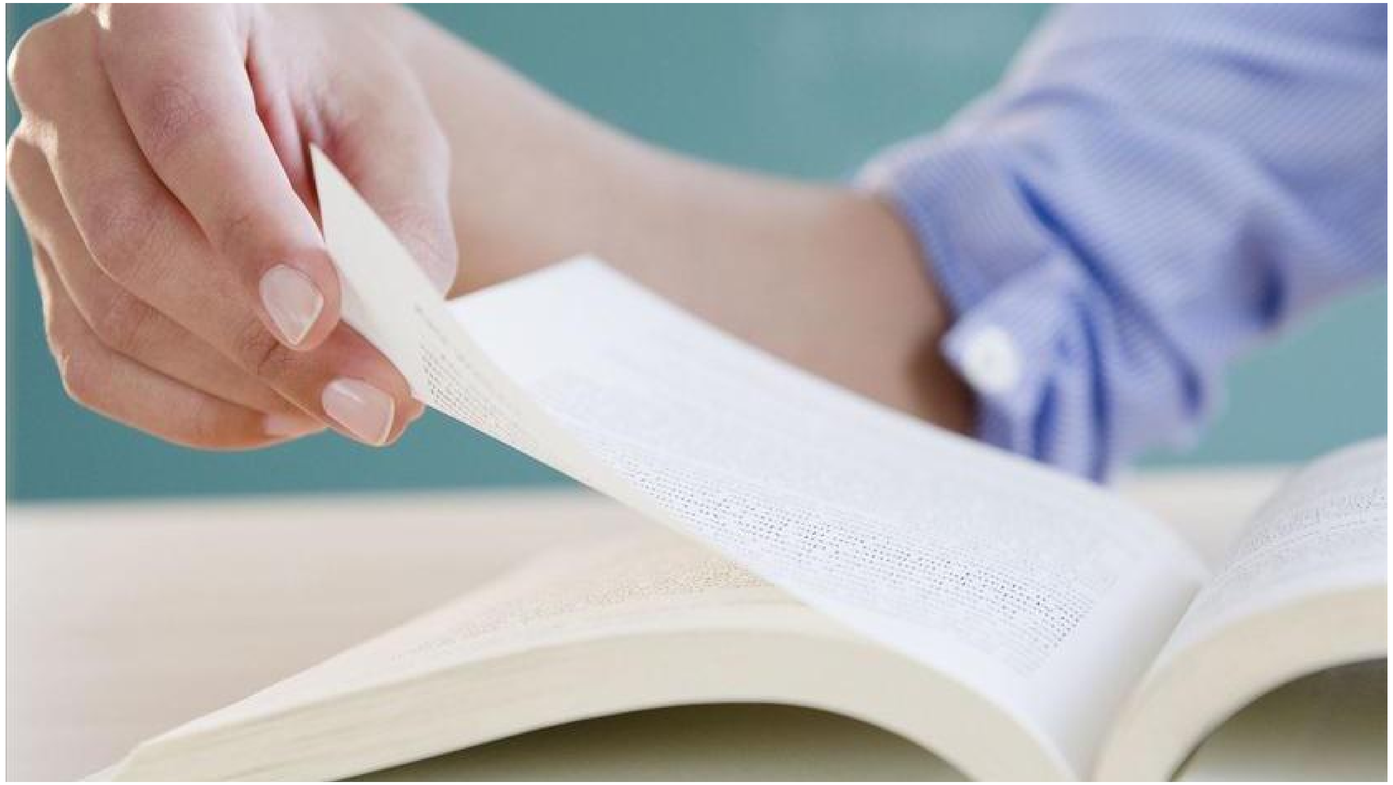
1.Go to the Run menu and select the Run option.
2. Select the Class name and click on Run.
3.right-click editor and elect Run 'HelloWorld.main()' in the popup



```java
package com.example.helloworld;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HelloWorld [C:\Users\jyh\IdeaProjects\HelloWorld] - ...\src\com\example
HelloWorld src com example helloworld HelloWorld
1: Project
Project
HelloWorld.java
HelloWorld C:\Users\jyh\IdeaProjects\HelloWorld
.idea
src
com.example.helloworld
HelloWorld
HelloWorld.iml
External Libraries
Scratches and Consoles
Show Context Actions Alt+Enter
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Paste from History... Ctrl+Shift+V
Paste without Formatting Ctrl+Alt+Shift+V
Column Selection Mode Alt+Shift+Insert
Find Usages Alt+F7
Refactor
Folding
Analyze
Go To
Generate... Alt+Insert
Run 'HelloWorld.main()' Ctrl+Shift+F10
Debug 'HelloWorld.main()'
Run 'HelloWorld.main()' with Coverage
Create 'HelloWorld.main()'...
Show in Explorer
File Path Ctrl+Alt+F12
Open in Terminal
Local History
Compare with Clipboard
Create Gist...
2: Favorites
7: Structure

2 Primitive types



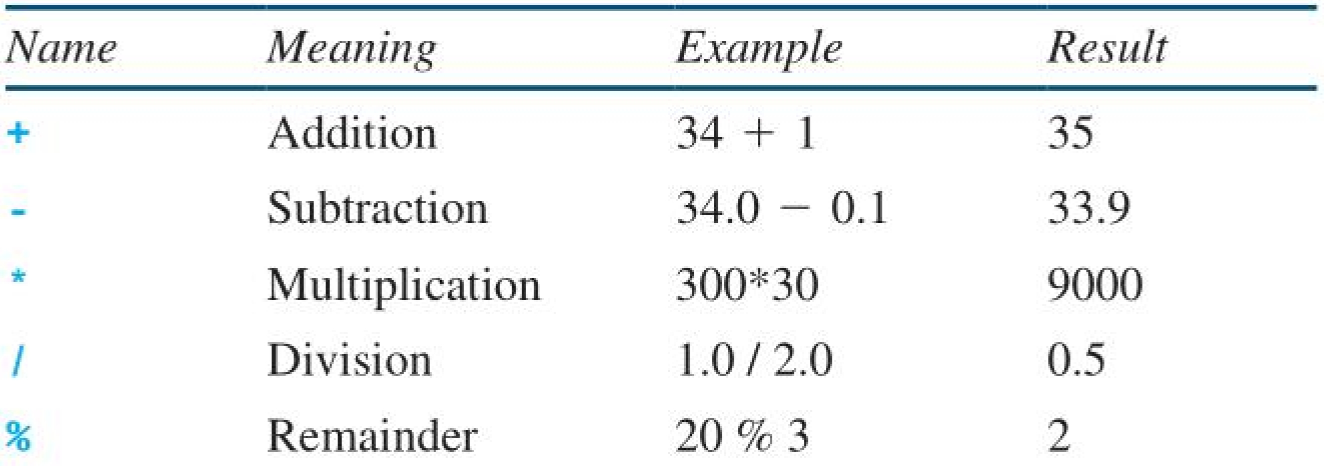
Data type and its range

Table:List of Java's primitive data types

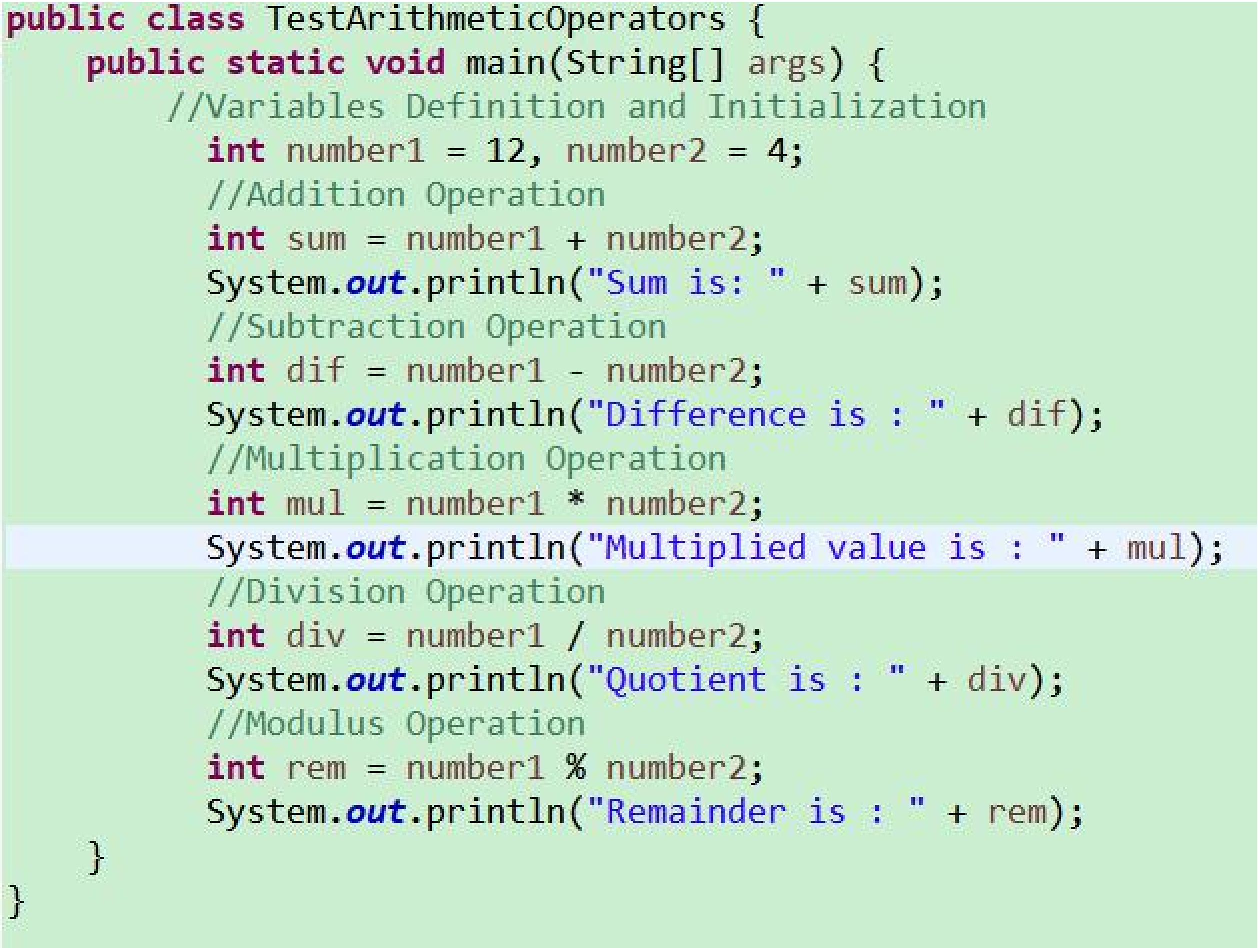
Type	Size in Bytes	Range
byte	1 byte	-2^7 to 2^7 -1 (-128 to 127)
short	2 bytes	-2^{15} to 2^{15} -1 (-32768 to 32767)
int	4 bytes	-2^{31} to 2^{31} -1 (-2147483648 to 2147483647)
long	8 bytes	-2^{63} to 2^{63} -1(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
float	4 bytes	approximately ±3.40282347E+38F (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

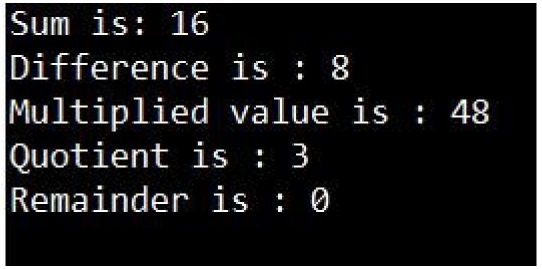
Data Operations(Basic Arithmetic Operators)



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



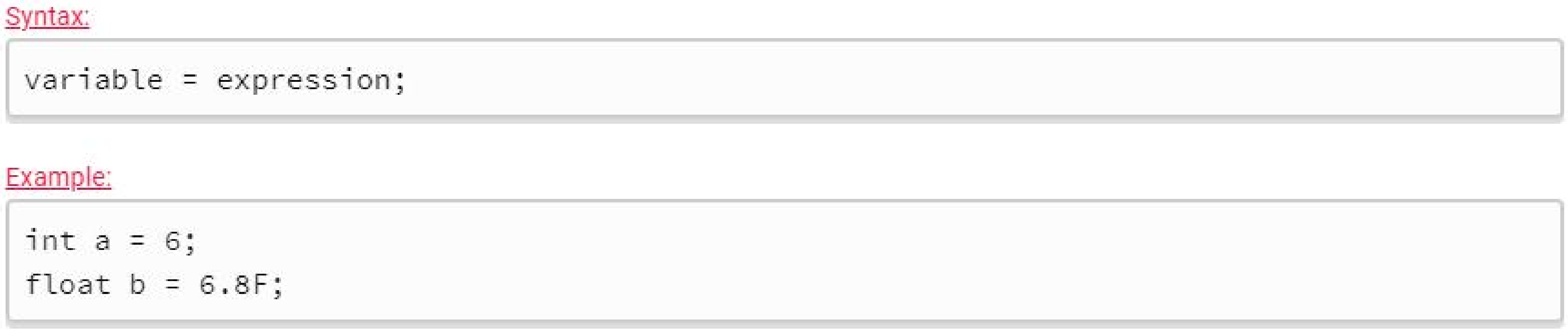
```java
public class TestArithmeticOperators {
    public static void main(String[] args) {
        //Variables Definition and Initialization
         int number1 = 12, number2 = 4;
         //Addition Operation
         int sum = number1 + number2;
         System.out.println("Sum is: " + sum);
         //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;
```

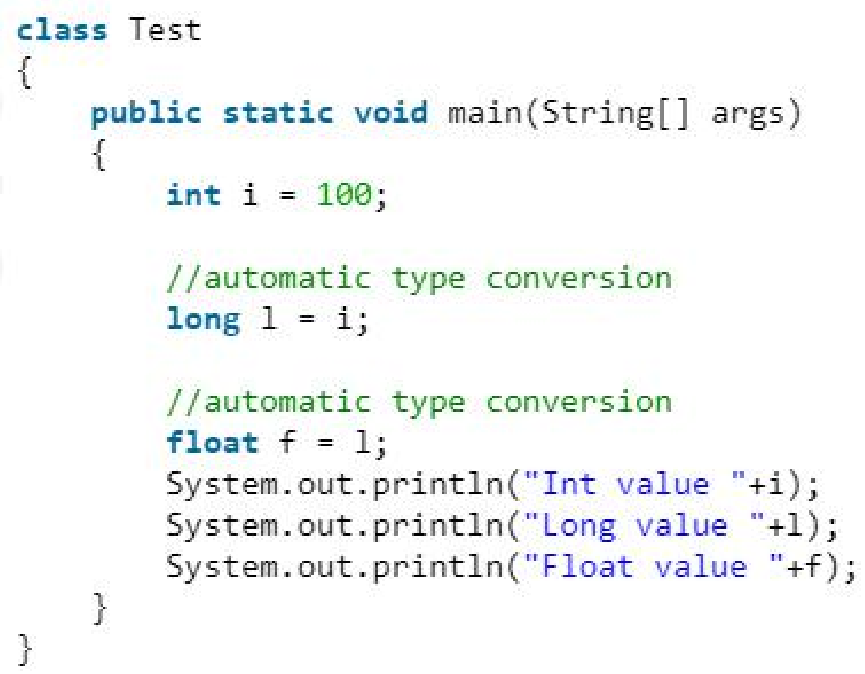
Data type Conversions

◆Widening or Automatic Type Conversion

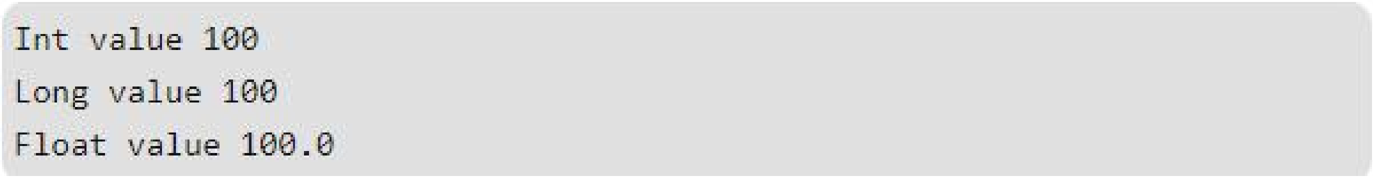
- The two data types are compatible.(char and boolean are not compatible with each other.)
- When we assign value of a smaller data type to a bigger data type.

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);
    }
}
```

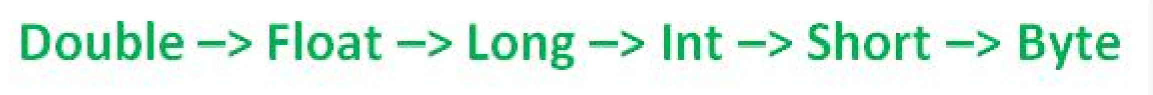


```
Int value 100
Long value 100
Float value 100.0
```

Data type Conversions

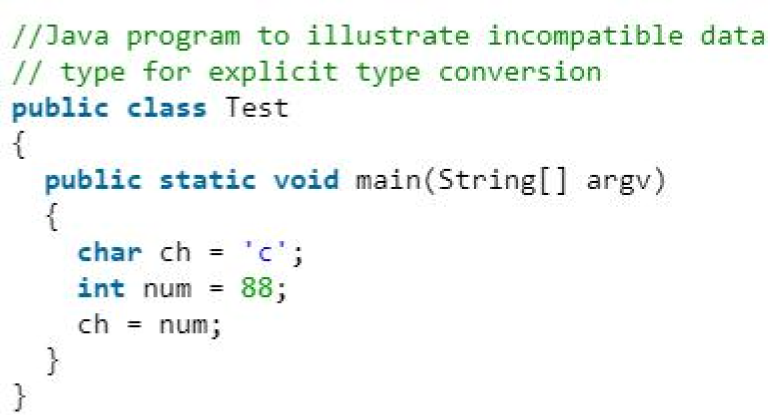
◆Narrowing or Explicit Conversion

- 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

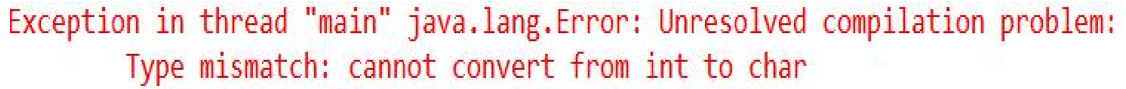


Double –> Float –> Long –> Int –> Short –> Byte

Example:

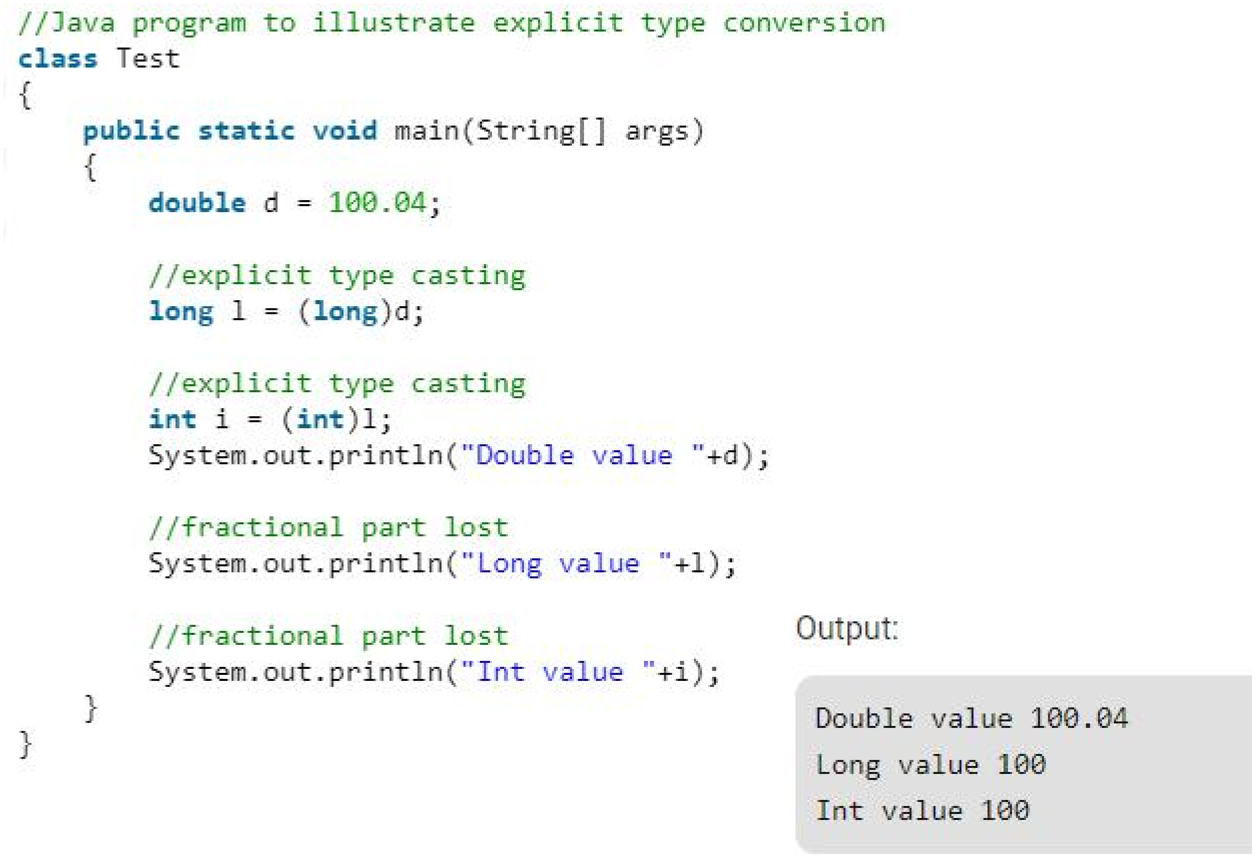


```java
//Java program to illustrate incompatible data
// type for explicit type conversion
public class Test
{
  public static void main(String[] argv)
  {
    char ch = 'c';
    int num = 88;
    ch = num;
  }
}
```



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

Example:



```java
//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
```

3 Naming rules of Java

贾艳红 Jana

Email:jiayh@mail.sustech.edu.cn

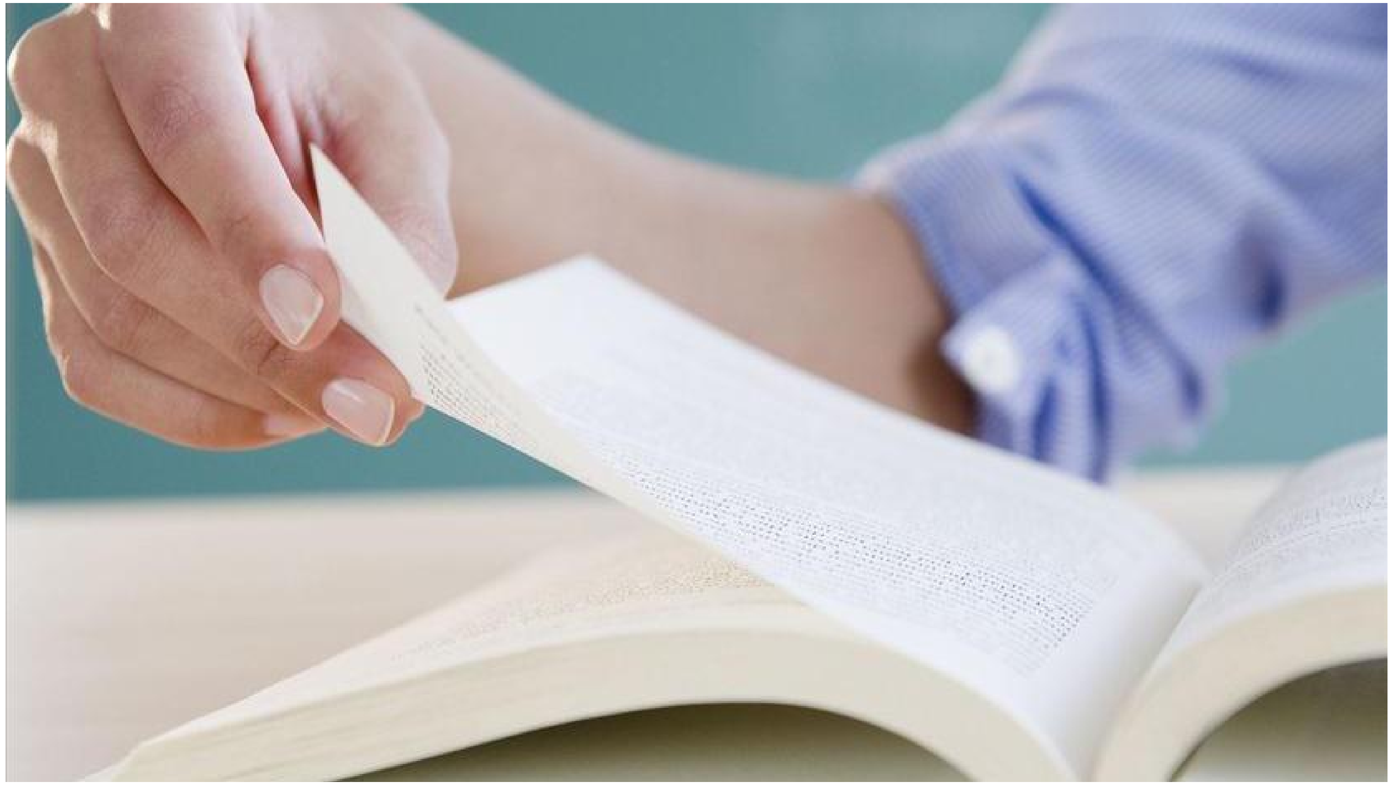
Naming rules of Java

◆**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.



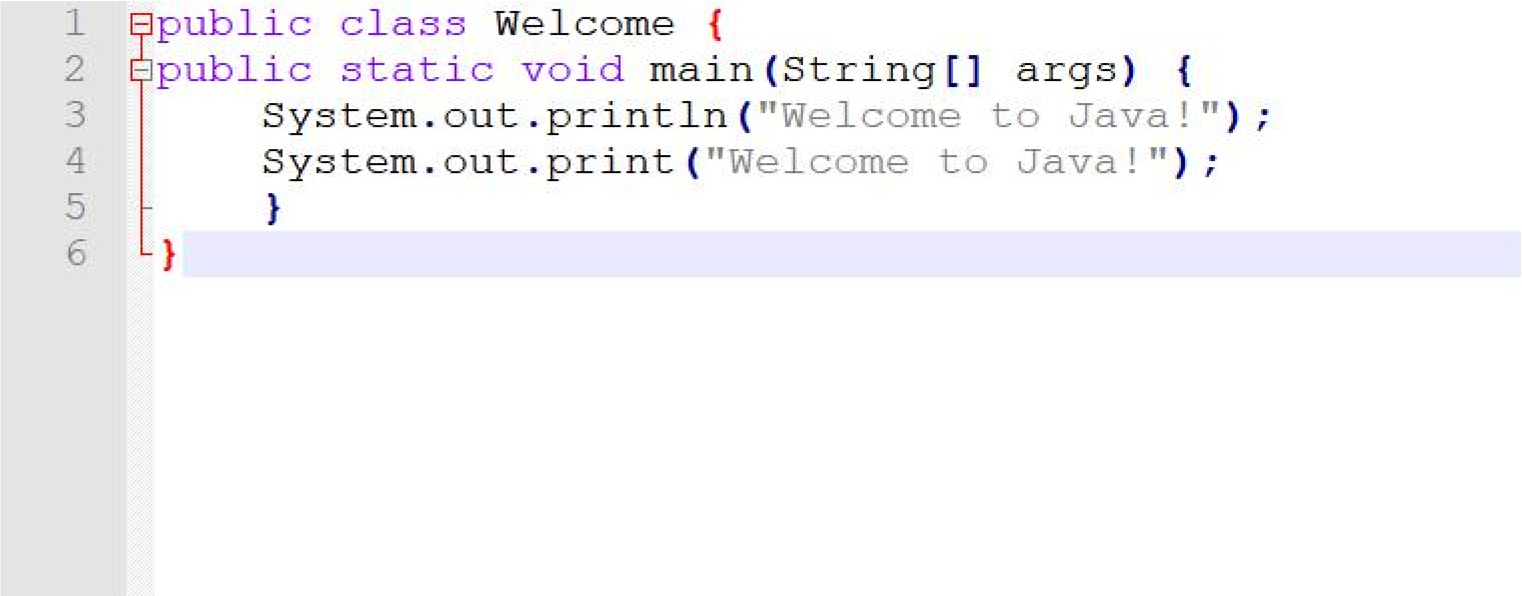
4 Output

- 3.1 Print without format
- 3.2 Print with format
- 3.3 Case Study

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

Print without format

- System.out.println();
- System.out.print();



```
public class Welcome {
public static void main(String[] args) {
    System.out.println("Welcome to Java!");
    System.out.print("Welcome to Java!");
    }
}
```

Print with format

- System.out.printf();
- System.out.format(); //usage similar with System.out.printf();

Table Frequently Used Format Specifiers

Format Specifier	Output	Example
%b	A Boolean value	True or false
%c	A character	‘a’
%d	A decimal integer	200
%f	A floating-point number	45.460000
%e	A number in standard scientific notation	4.556000e+01
%s	A string	“Java is cool”

Print with format

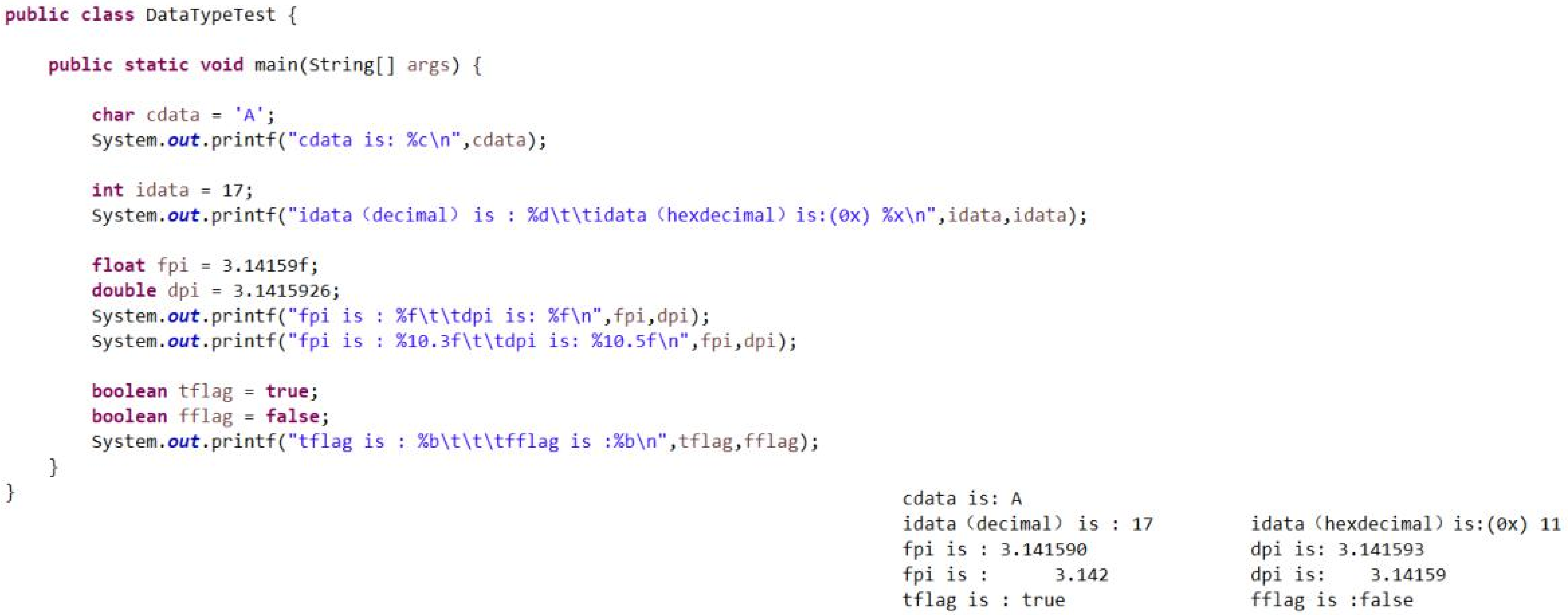
Table Examples of Specifying Width and Precision

Example	Output
%5c	Output the character and add **four spaces** before **the character item**, because **the width is 5**.
%6b	Output the Boolean value and add **one space** before the **false** value and **two spaces** before the **true** value.
%5d	Output the integer item with **width 5.** If the number of digits in the item is <5, **add spaces before the number**. If the number of digits in the item is >5, **the width is automatically increased**.
%10.2f	Output the floating-point item with **width 10 including a decimal point and two digits after the point**. Thus, there are seven digits allocated before the decimal point. If the number of digits before the decimal point in the item is **<7, add spaces before the number**. If the number of digits before the decimal point in the item is **> 7, the width is automatically increased**.
%10.2e	Output the floating-point item with width 10 including a decimal point, two digits after the point and the exponent part. If the displayed number in scientific notation has width 6 10, add spaces before the number.
%12s	Output the string with width 12 characters. If the string item has fewer than 12 characters, add spaces before the string. If the string item has more than 12 characters, the width is automatically increased.

```
System.out.printf("%3d#%2s#%4.2f\n", 1234, "Java", 51.6653);
```

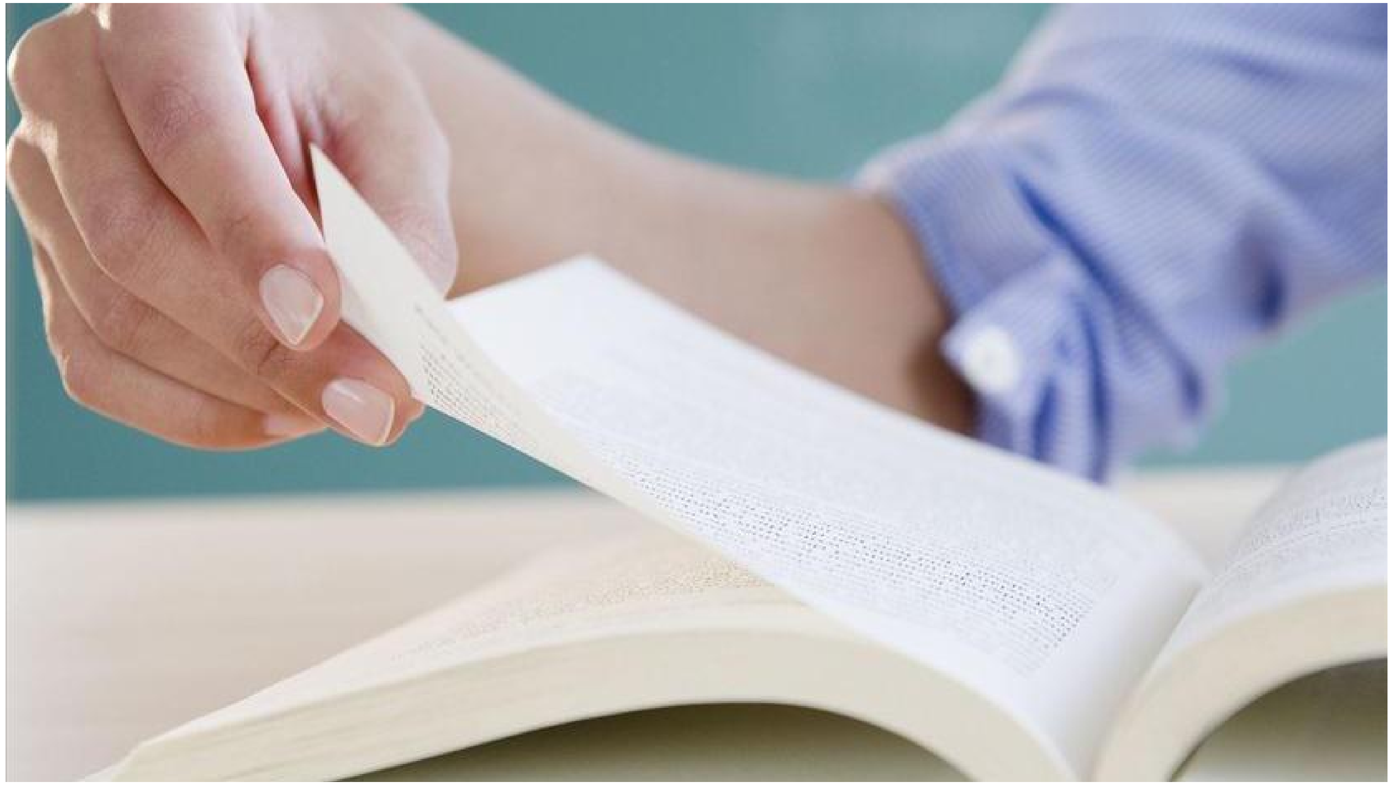
Output: 1234#Java#51.67

Case Study



```java
public class DataTypeTest {

    public static void main(String[] args) {

        char cdata = 'A';
        System.out.printf("cdata is: %c\n",cdata);

        int idata = 17;
        System.out.printf("idata（decimal） is : %d\t\tidata（hexdecimal） is:(0x) %x\n",idata,idata);

        float fpi = 3.14159f;
        double dpi = 3.1415926;
        System.out.printf("fpi is : %f\t\tdpi is: %f\n",fpi,dpi);
        System.out.printf("fpi is : %10.3f\t\tdpi is: %10.5f\n",fpi,dpi);

        boolean tflag = true;
        boolean fflag = false;
        System.out.printf("tflag is : %b\t\t\tfflag is :%b\n",tflag,fflag);
    }
}
```

```
cdata is: A
idata（decimal） is : 17         idata（hexdecimal） is:(0x) 11
fpi is : 3.141590               dpi is: 3.141593
fpi is :      3.142             dpi is:    3.14159
tflag is : true                 fflag is :false
```



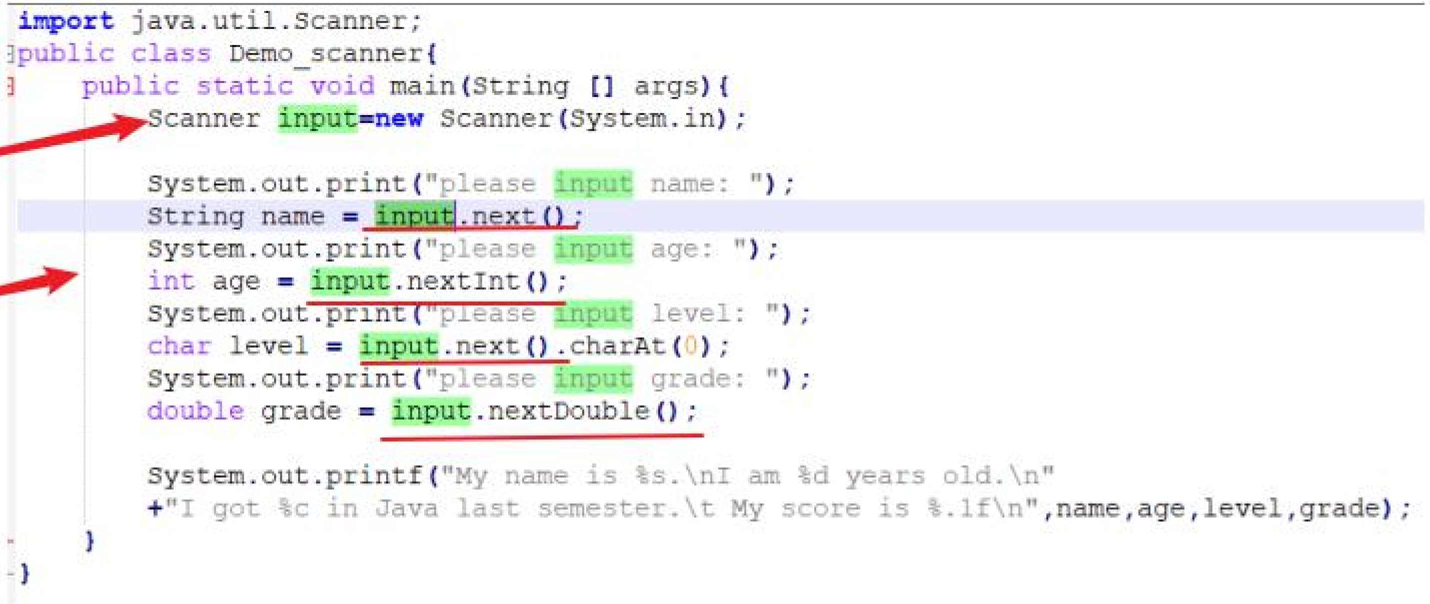
5 Input

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

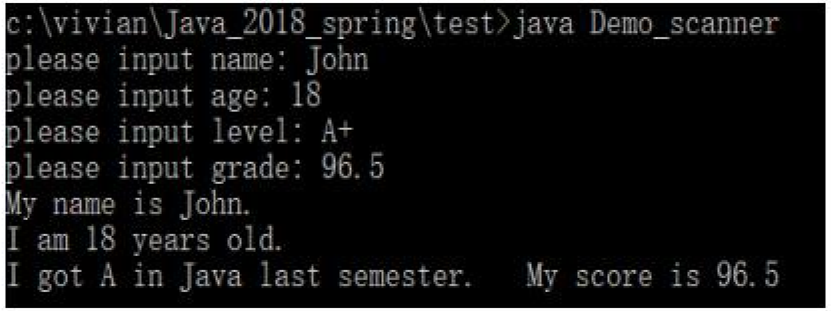
Using Scanner to get input

In you code

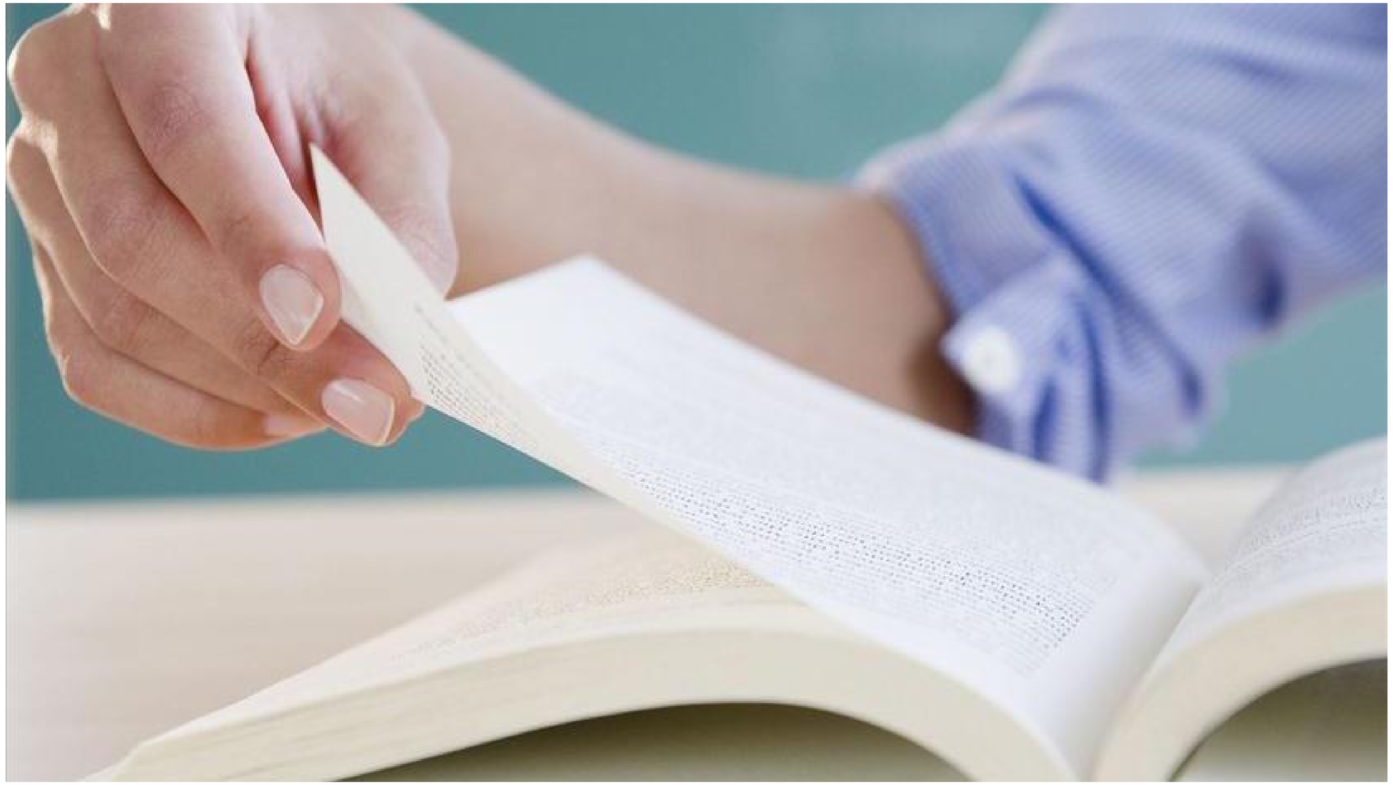
- 1. **import java.util.Scanner**
- 2. **new an Scanner object**
- 3. invoke the **method** of the Scanner object to get input data
    - 3-1: next() to get a string data
    - 3-2: nextInt() to get an integer data
    - 3-3: nextDouble() to get a double data
- 4. if all the input process is end, invoke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



3 Exercises

Complete the exercises in the **2020S-Java-A-Lab-2.pdf** and submit to the blackboard as required.

THANK YOU

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