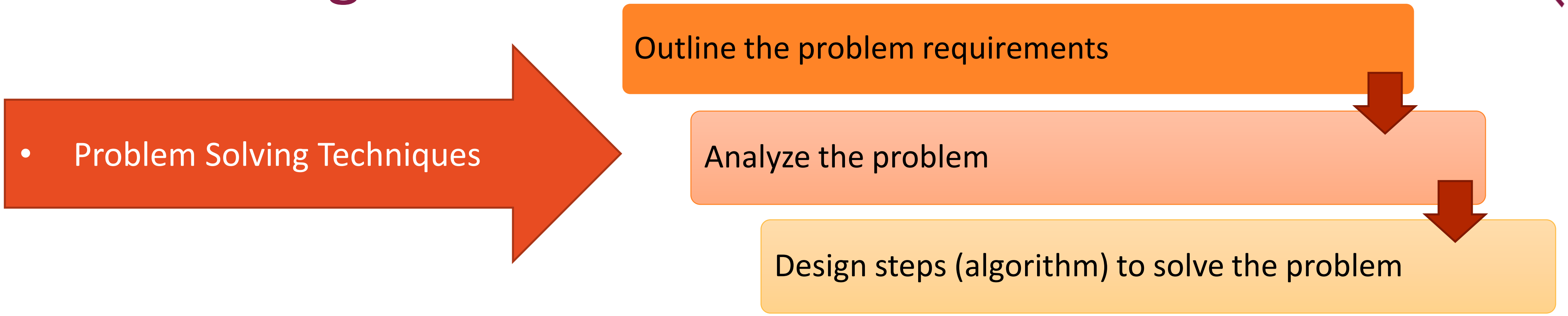
Introduction to JAVA Programming Language

Computer Engineering

Yusramohammed@tiu.edu.iq

2022 - 2023

Problem Solving Process



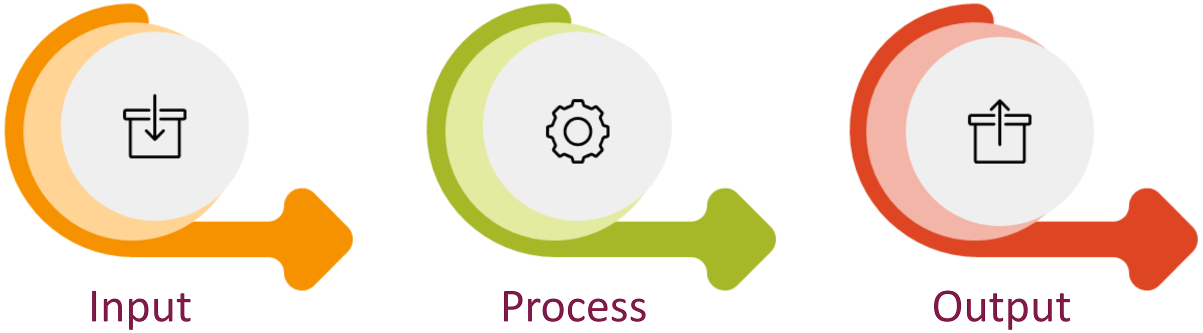
- Problem Solving Techniques

Outline the problem requirements

Analyze the problem

Design steps (algorithm) to solve the problem

- Three steps that a program typically performs:



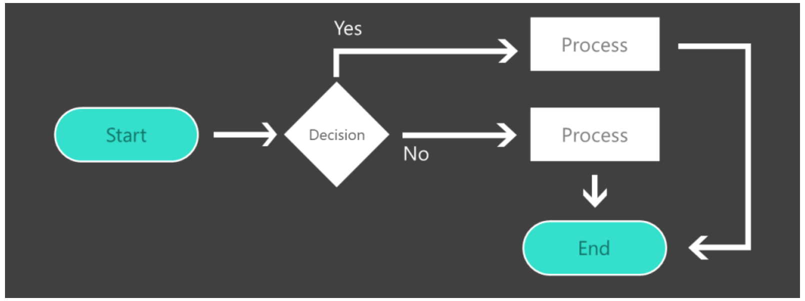
- Gather input
  - from keyboard
  - from files on disk drives

Process the input

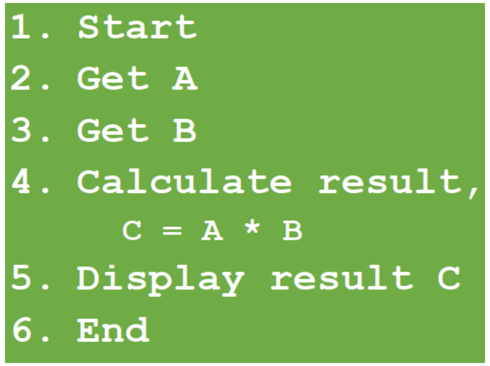
- Display the result as output
  - send it to the screen
  - write to a file

problem solving methods

Flowchart

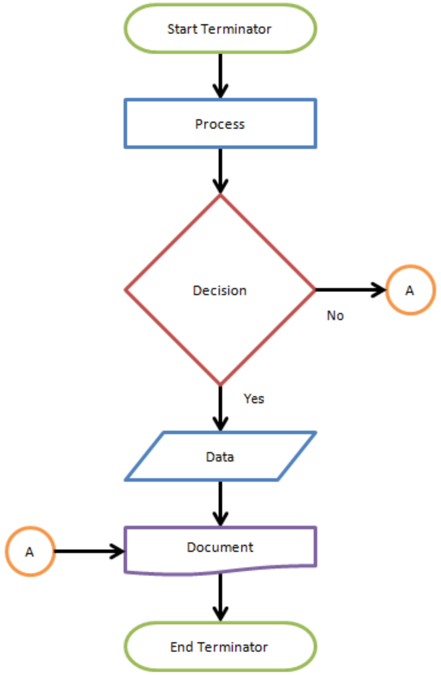


Pseudo-code



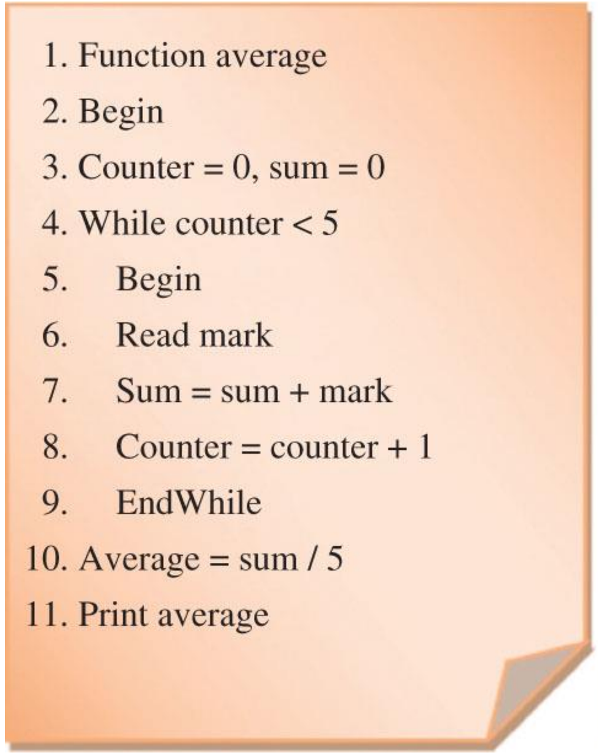
```
1. Start
2. Get A
3. Get B
4. Calculate result,
     C = A * B
5. Display result C
6. End
```

Structured Chart

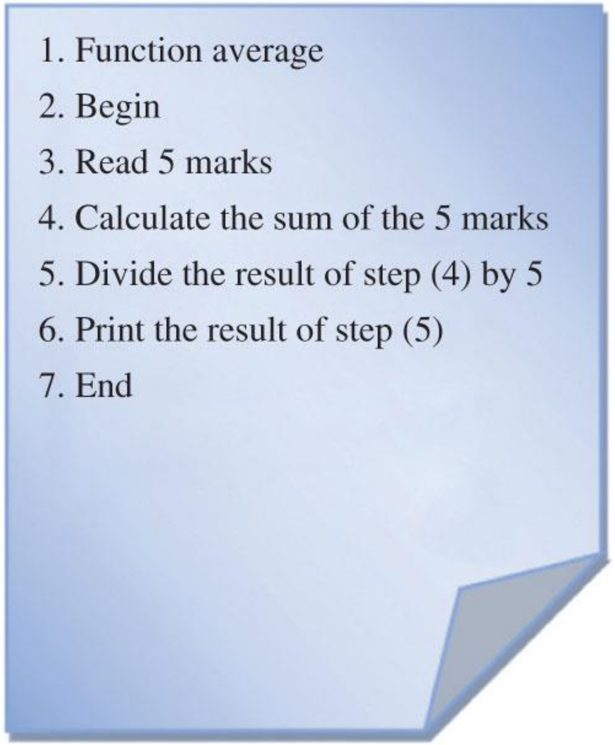


Pseudo-Code

- Pseudocode is a semiformal, English-like language with limited vocabulary that can be used to design & describe algorithms.
- Purpose- to define the procedural logic of an algorithm
- in a simple, easy-to-understand for its readers.
- Free of syntactical complications of programming language.
- **Example**: Find the average of 5 marks entered by user.



```
Function average
Begin
Counter = 0, sum = 0
While counter < 5
    Begin
    Read mark
    Sum = sum + mark
    Counter = counter + 1
    EndWhile
Average = sum / 5
Print average
```



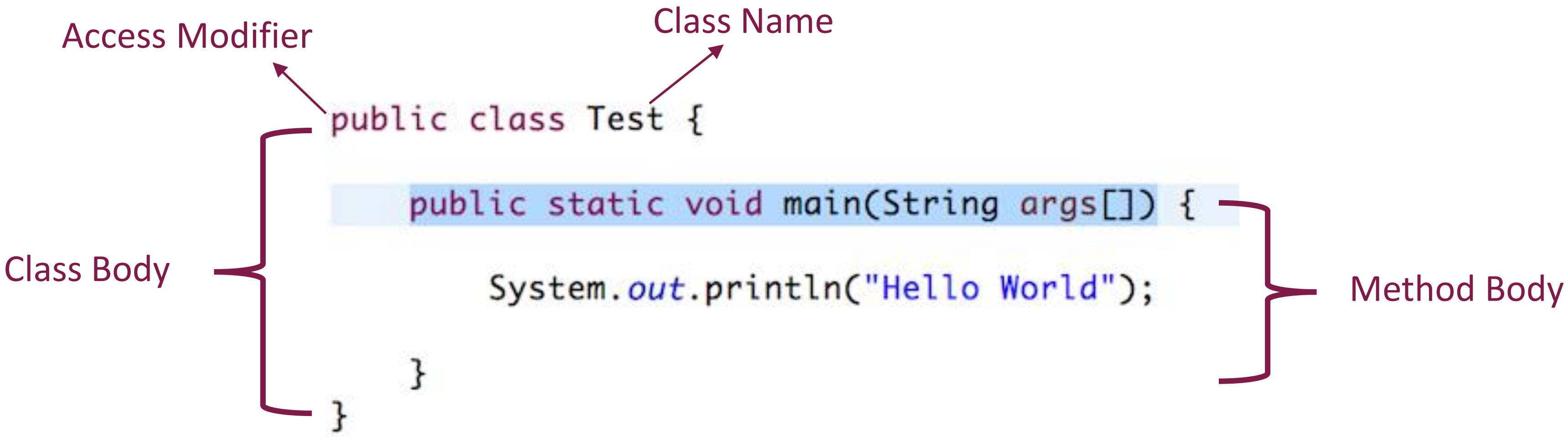
```
1. Function average
2. Begin
3. Read 5 marks
4. Calculate the sum of the 5 marks
5. Divide the result of step (4) by 5
6. Print the result of step (5)
7. End
```

Introduction to Java Application

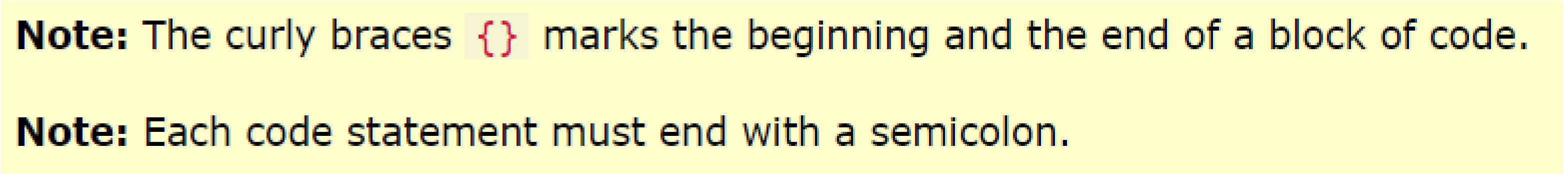
- High-level language programs are usually written (coded) as ASCII text into a source code file. A unique file extension (Examples: .asm .c .cpp .java .js .py) is used to identify it as a source code file.
- By convention, class names begin with a capital letter and capitalize the first letter of each word they include (e.g., SampleClassName).
- Java is ***case sensitive***—uppercase and lowercase letters are distinct—so value and Value are different (but both valid) identifiers.
- A class name is an identifier—a series of characters consisting of **letters**, **digits**, **underscores (_)** and **dollar signs ($)** that does not begin with a digit and does not contain spaces.
- Some valid identifiers are Welcome1, $value,_value, m_inputField1 and button7.

Introduction to Java Application

- A Java application is a computer program that executes when you use the **java** command to launch the Java Virtual Machine (JVM).
- To write and run a java code for an application, every line of code must be enclosed in a ***class***.
- Java class declarations normally contain one or more methods. For a Java application, one of the methods must be called ***main*** and must be defined as



```
public class Test {

    public static void main(String args[]) {

        System.out.println("Hello World");

    }
}
```

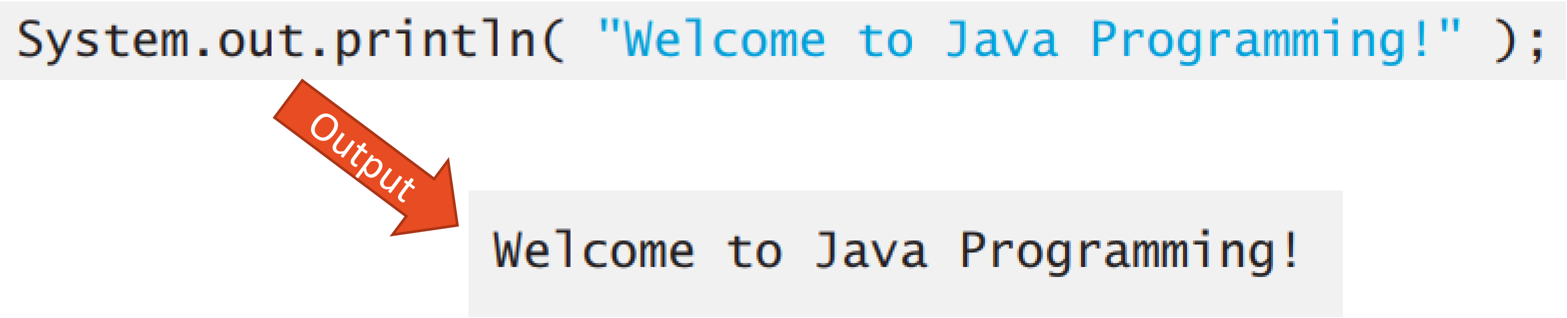


Note: The curly braces `{}` marks the beginning and the end of a block of code.

Note: Each code statement must end with a semicolon.

Display messages in java programs

- **System.out.println();** → We use this sentence to show messages and information in the output command window.
- **println()**→ Print the line of code and go to the next line.
- **“ ”**→ Any thing written inside the double quotation will be shown in the output.
- Every code of statement should be ended with semicolon **;**

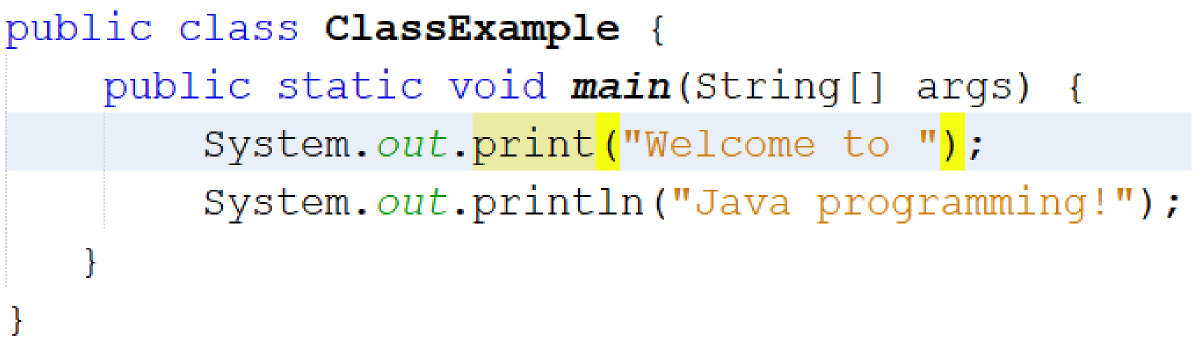


```
System.out.println( "Welcome to Java Programming!" );
```

```
Welcome to Java Programming!
```

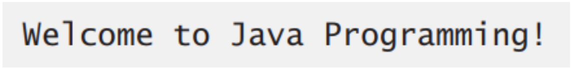
Display messages using print()

- **print()→** print the line of code and remain in the same line.

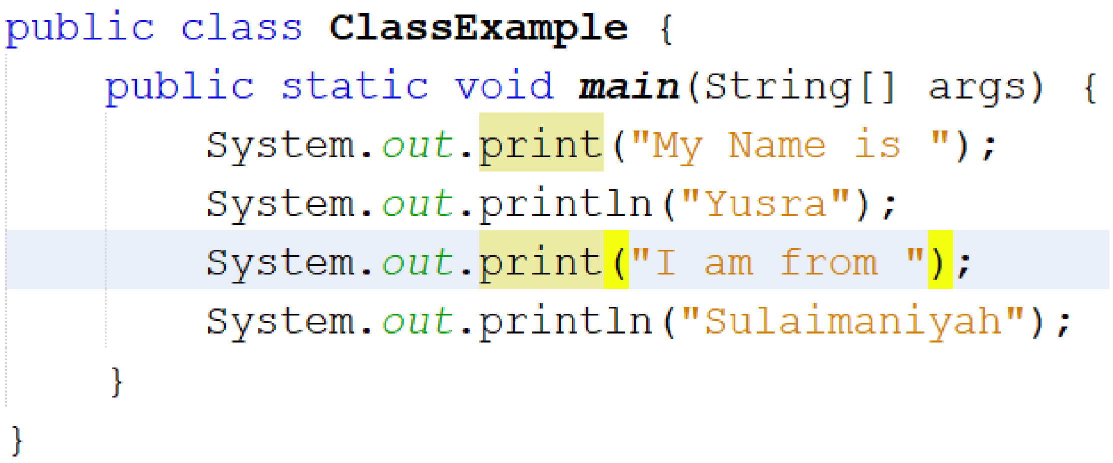


```
public class ClassExample {
    public static void main(String[] args) {
        System.out.print("Welcome to ");
        System.out.println("Java programming!");
    }
}
```

Output

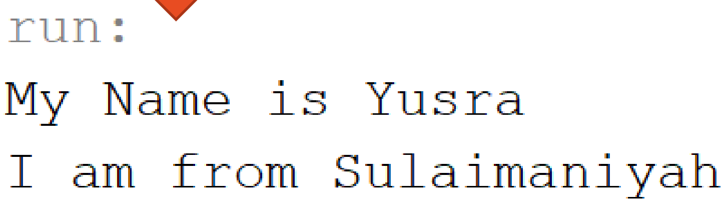


```
Welcome to Java Programming!
```



```
public class ClassExample {
    public static void main(String[] args) {
        System.out.print("My Name is ");
        System.out.println("Yusra");
        System.out.print("I am from ");
        System.out.println("Sulaimaniyah");
    }
}
```

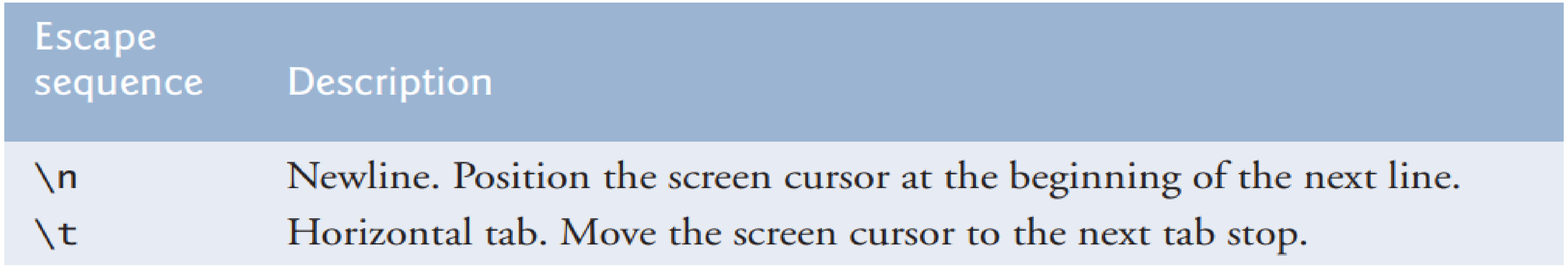
Output



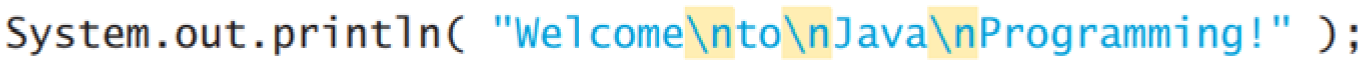
```
run:
My Name is Yusra
I am from Sulaimaniyah
```

Display messages in java programs

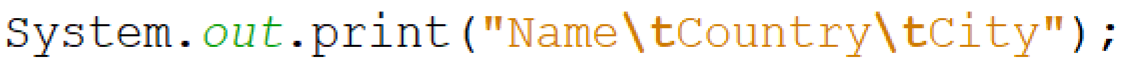
- Escape sequences are used in the “ ” of the System.out.print() sentence. The table shows two types of Escape sequences.



Escape sequence	Description
\n	Newline. Position the screen cursor at the beginning of the next line.
\t	Horizontal tab. Move the screen cursor to the next tab stop.

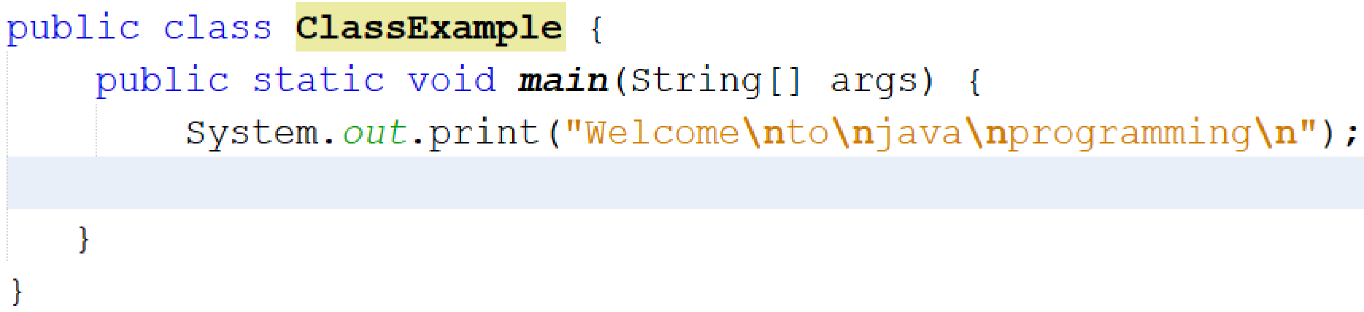


```
System.out.println( "Welcome\nto\nJava\nProgramming!" );
```



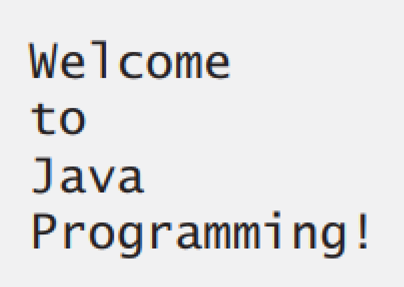
```
System.out.print("Name\tCountry\tCity");
```

Display messages in java programs

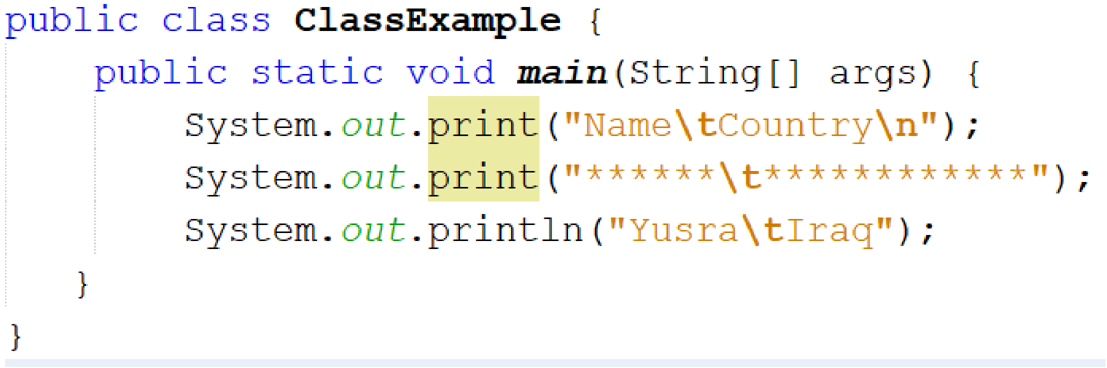


```java
public class ClassExample {
    public static void main(String[] args) {
        System.out.print("Welcome\nto\njava\nprogramming\n");

    }
}
```

Output

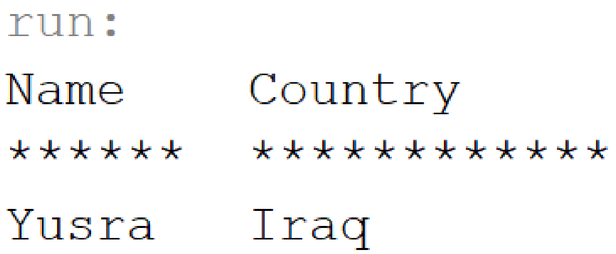


```
Welcome
to
Java
Programming!
```



```java
public class ClassExample {
    public static void main(String[] args) {
        System.out.print("Name\tCountry\n");
        System.out.print("******\t************");
        System.out.println("Yusra\tIraq");
    }
}
```

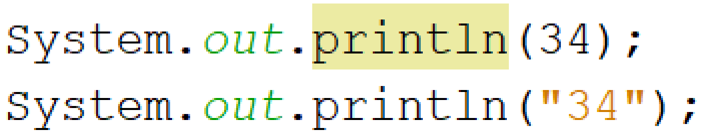
Output



```
run:
Name    Country
******  ************
Yusra   Iraq
```

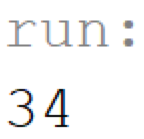
Output Numbers

- We can write numbers in double quotation or just write it without double quotation.



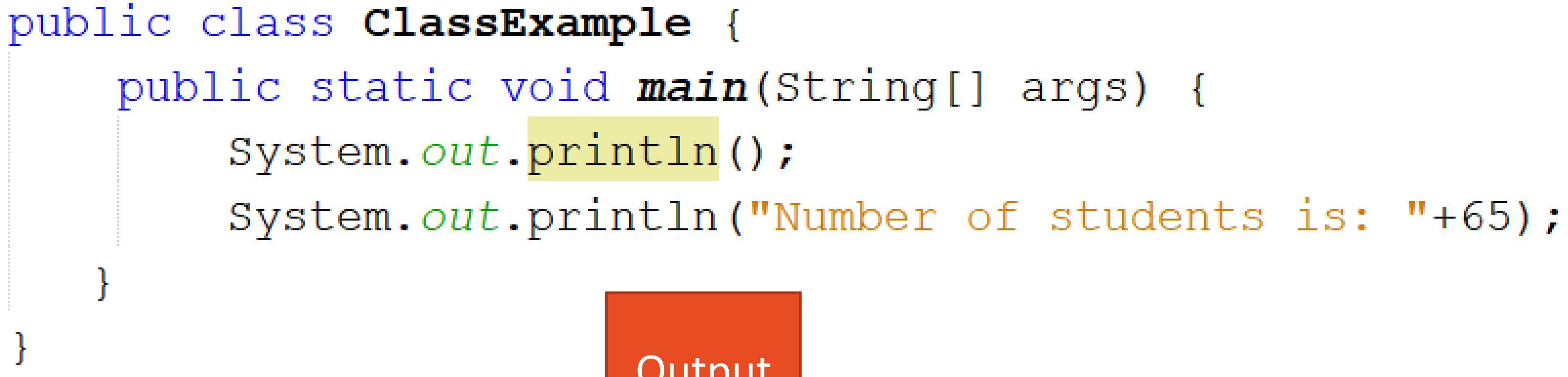
```
System.out.println(34);
System.out.println("34");
```

Both gives the same output



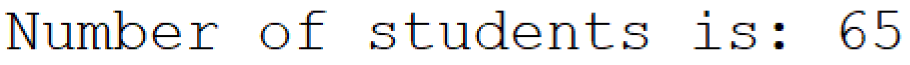
```
run:
34
```

- To combine a number with a text inside a double quotation we use + sign



```
public class ClassExample {
    public static void main(String[] args) {
        System.out.println();
        System.out.println("Number of students is: "+65);
    }
}
```

Output



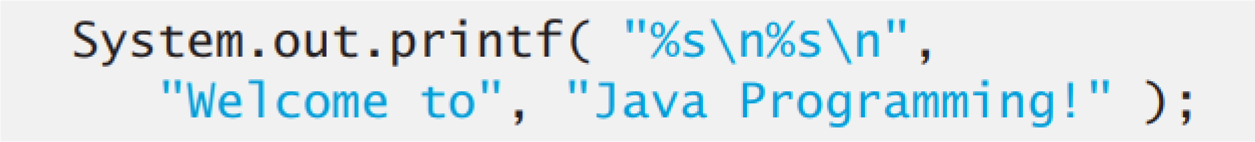
```
Number of students is: 65
```

Display messages using Printf()

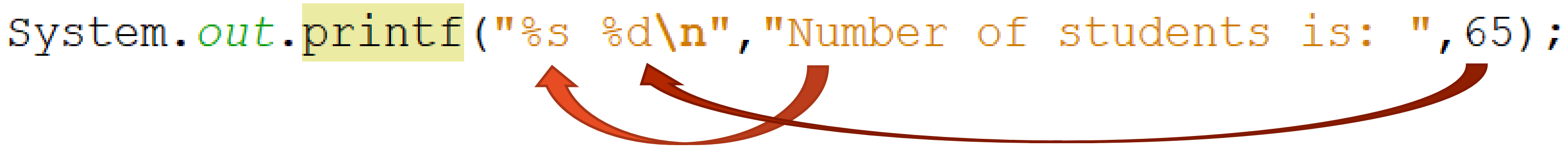
- **printf()→** can be used to print numbers and text on the output window. Based on the message to be shown in the output window, we use:

✓ %d→ for Numbers

✓ %s → for texts

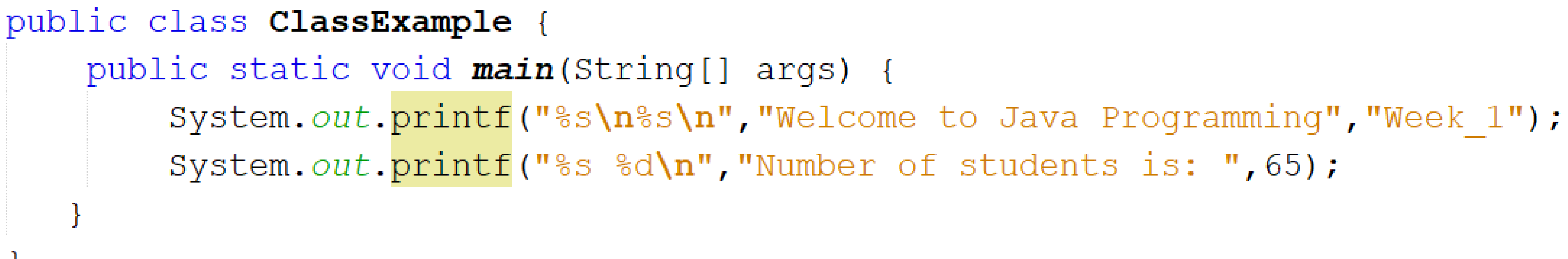


```
System.out.printf( "%s\n%s\n",
   "Welcome to", "Java Programming!" );
```



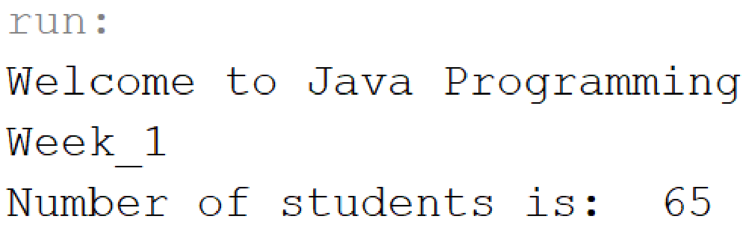
```
System.out.printf("%s %d\n","Number of students is: ",65);
```

Example:



```
public class ClassExample {
    public static void main(String[] args) {
        System.out.printf("%s\n%s\n","Welcome to Java Programming","Week_1");
        System.out.printf("%s %d\n","Number of students is: ",65);
    }
}
```

output



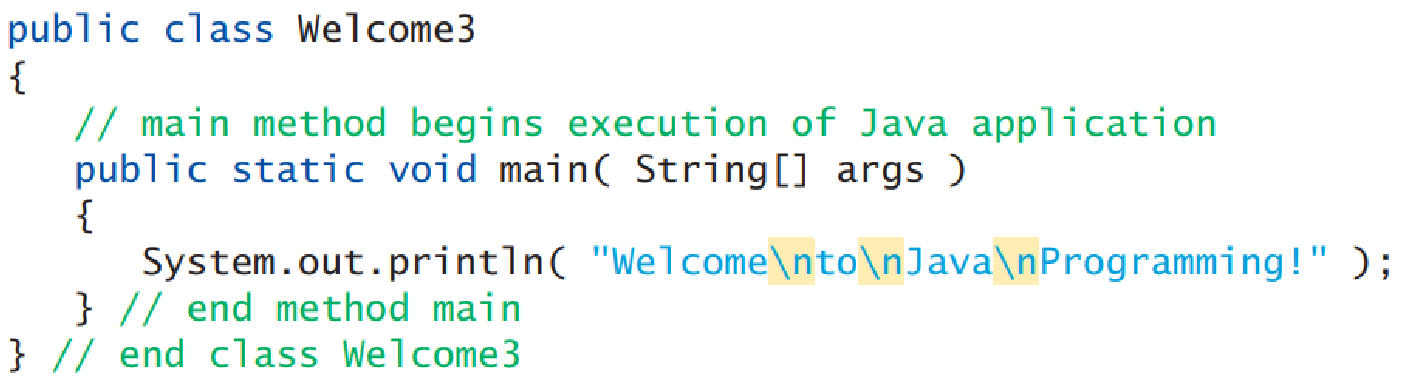
```
run:
Welcome to Java Programming
Week_1
Number of students is:  65
```

Comments in Java

- Comments can be used to explain Java code, and to make it more readable. It can also be used to prevent execution when testing alternative code.
- // → It is used for single-line comment

```java
// This is a comment
System.out.println("Hello World");
```

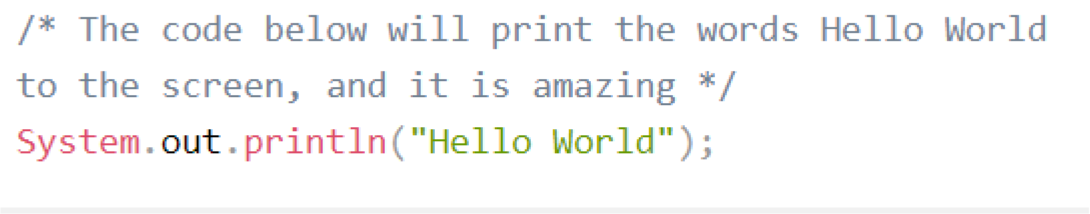
```java
System.out.println("Hello World"); // This is a comment
```



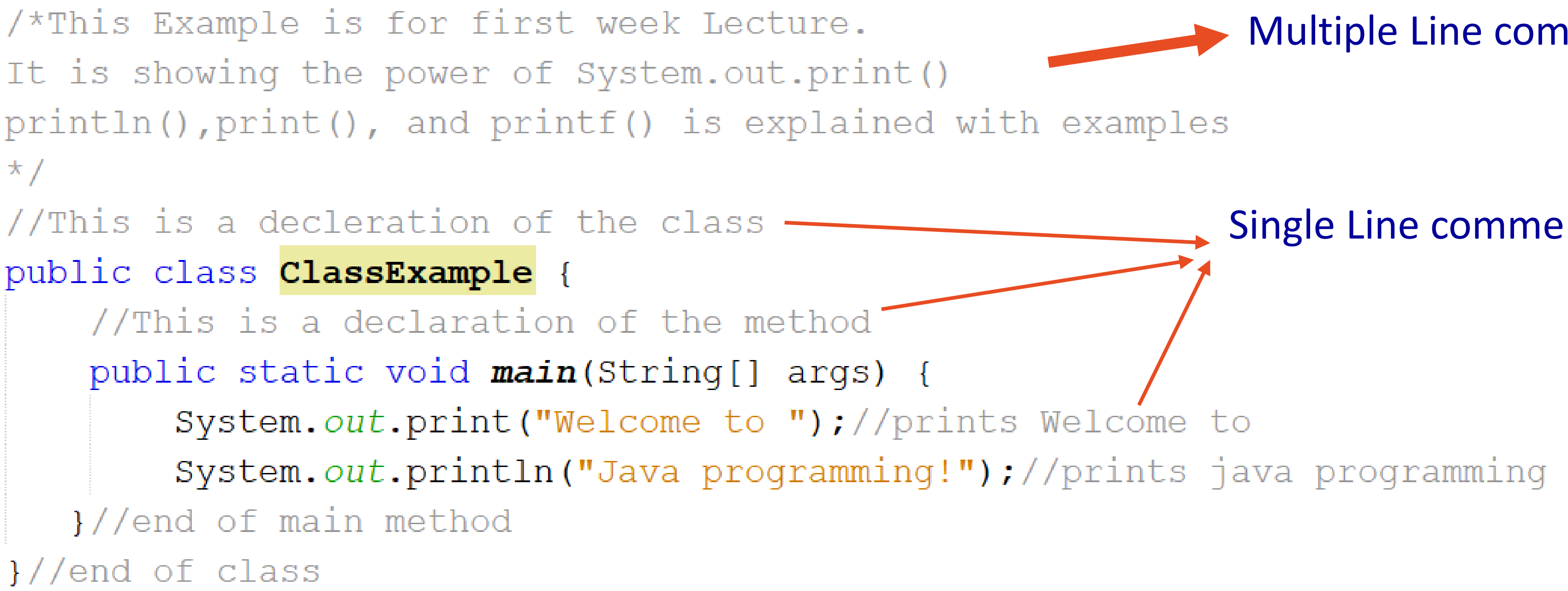
```java
public class Welcome3
{
   // main method begins execution of Java application
   public static void main( String[] args )
   {
      System.out.println( "Welcome\nto\nJava\nProgramming!" );
   } // end method main
} // end class Welcome3
```

Comments in Java

- /* */ → It is used for multiple-line comments
- Any text between /* and */ will be ignored by Java.



```
/* The code below will print the words Hello World
to the screen, and it is amazing */
System.out.println("Hello World");
```



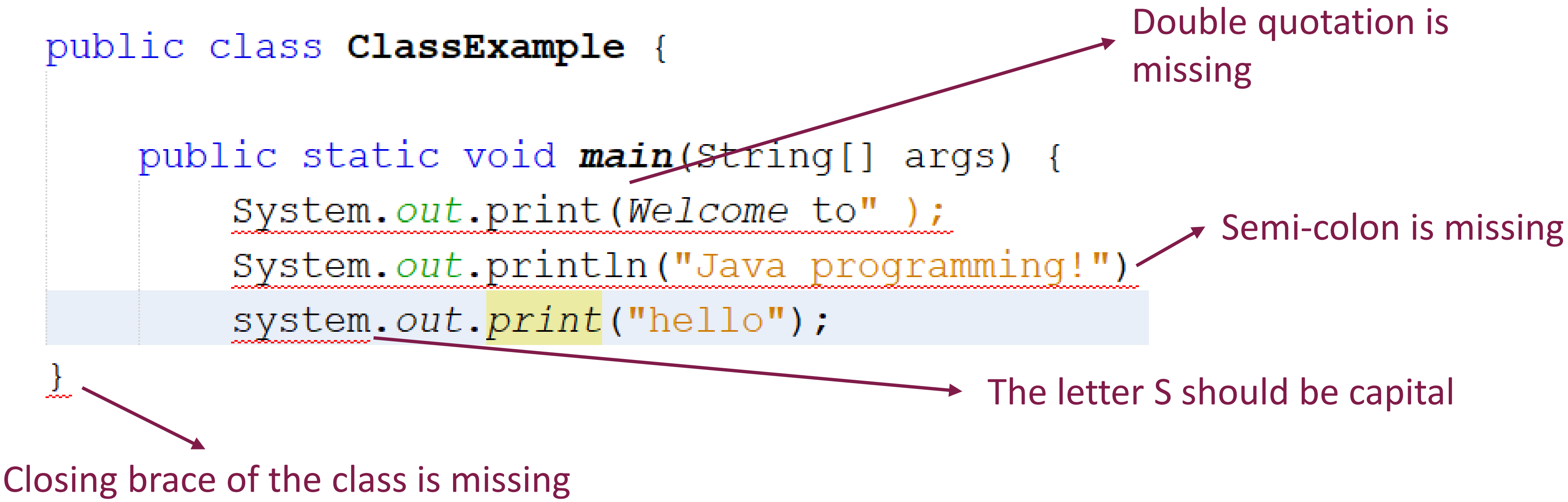
```
/*This Example is for first week Lecture.
It is showing the power of System.out.print()
println(),print(), and printf() is explained with examples
*/
//This is a decleration of the class
public class ClassExample {
    //This is a declaration of the method
    public static void main(String[] args) {
        System.out.print("Welcome to ");//prints Welcome to
        System.out.println("Java programming!");//prints java programming
   }//end of main method
}//end of class
```

Multiple Line comment

Single Line comment

Syntax Errors in Java

- Any line of code written in non-proper way is called syntax error.
- Examples of syntax error
- Forgetting to open or close the curly braces { } of the class or main method
- Forgetting to put semi-colon at the end of each sentence in the main method.
- Misspelling the codes in the main method, it means write the uppercase letters in lowercase and vise versa.



```
public class ClassExample {

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

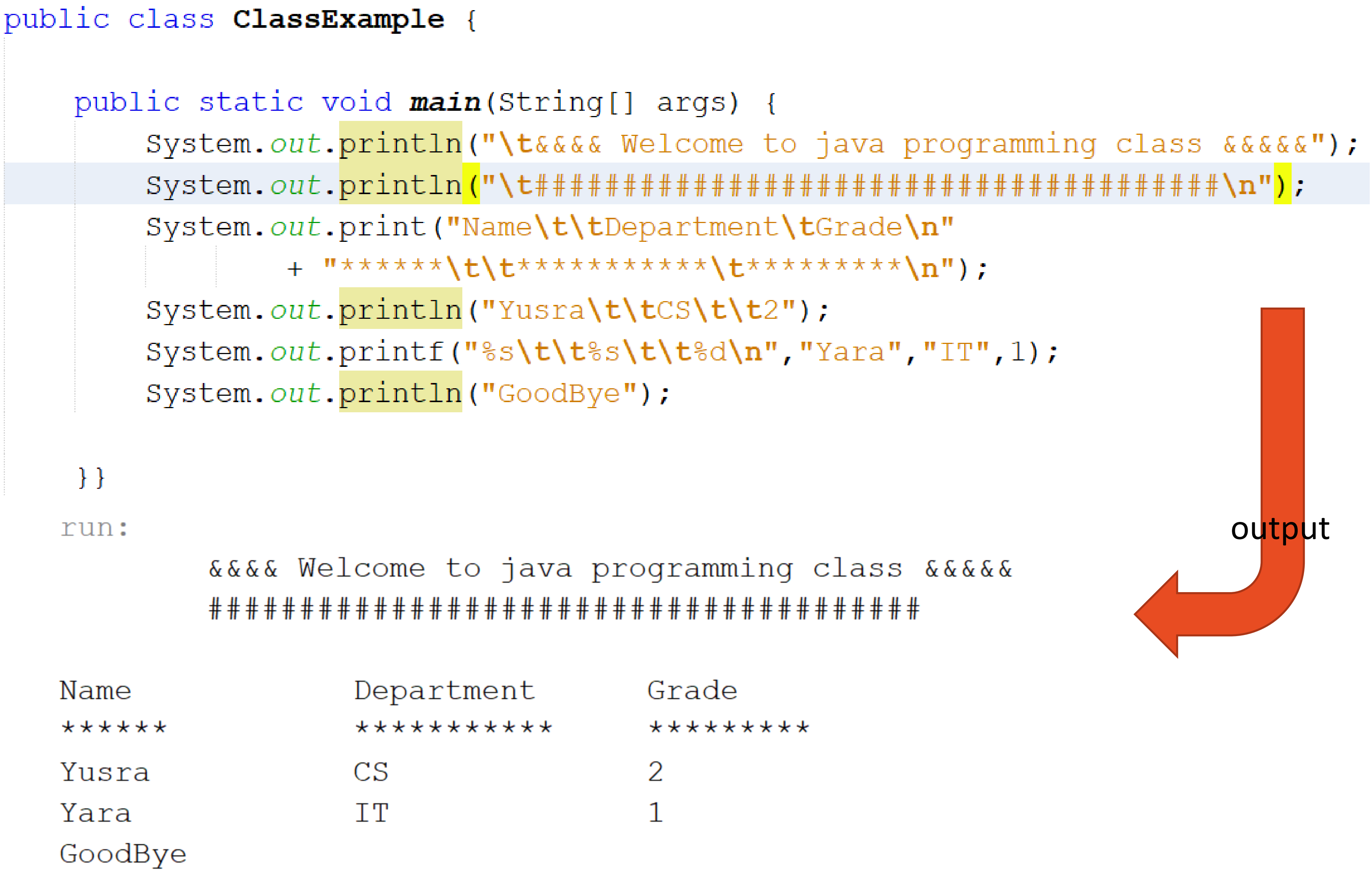
Double quotation is missing

Semi-colon is missing

The letter S should be capital

Closing brace of the class is missing

Sample Development



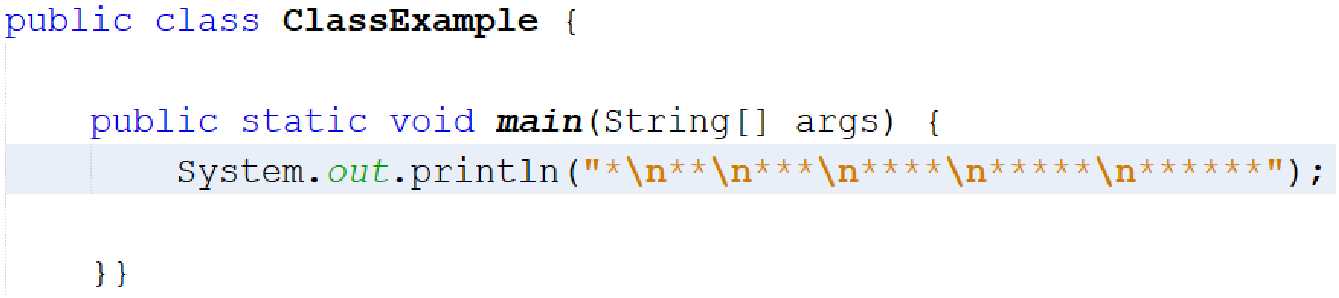
```java
public class ClassExample {

    public static void main(String[] args) {
        System.out.println("\t&&&& Welcome to java programming class &&&&&");
        System.out.println("\t#######################################\n");
        System.out.print("Name\t\tDepartment\tGrade\n"
                + "******\t\t***********\t*********\n");
        System.out.println("Yusra\t\tCS\t\t2");
        System.out.printf("%s\t\t%s\t\t%d\n","Yara","IT",1);
        System.out.println("GoodBye");

    }}
```

output

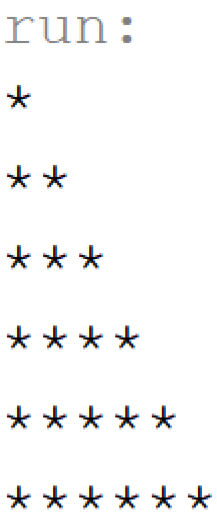
```
run:
        &&&& Welcome to java programming class &&&&&
        #######################################

Name            Department      Grade
******          ***********     *********
Yusra           CS              2
Yara            IT              1
GoodBye
```

Sample Development

- Write java program to print the triangle of asterisks using one System.out.println() line of code.



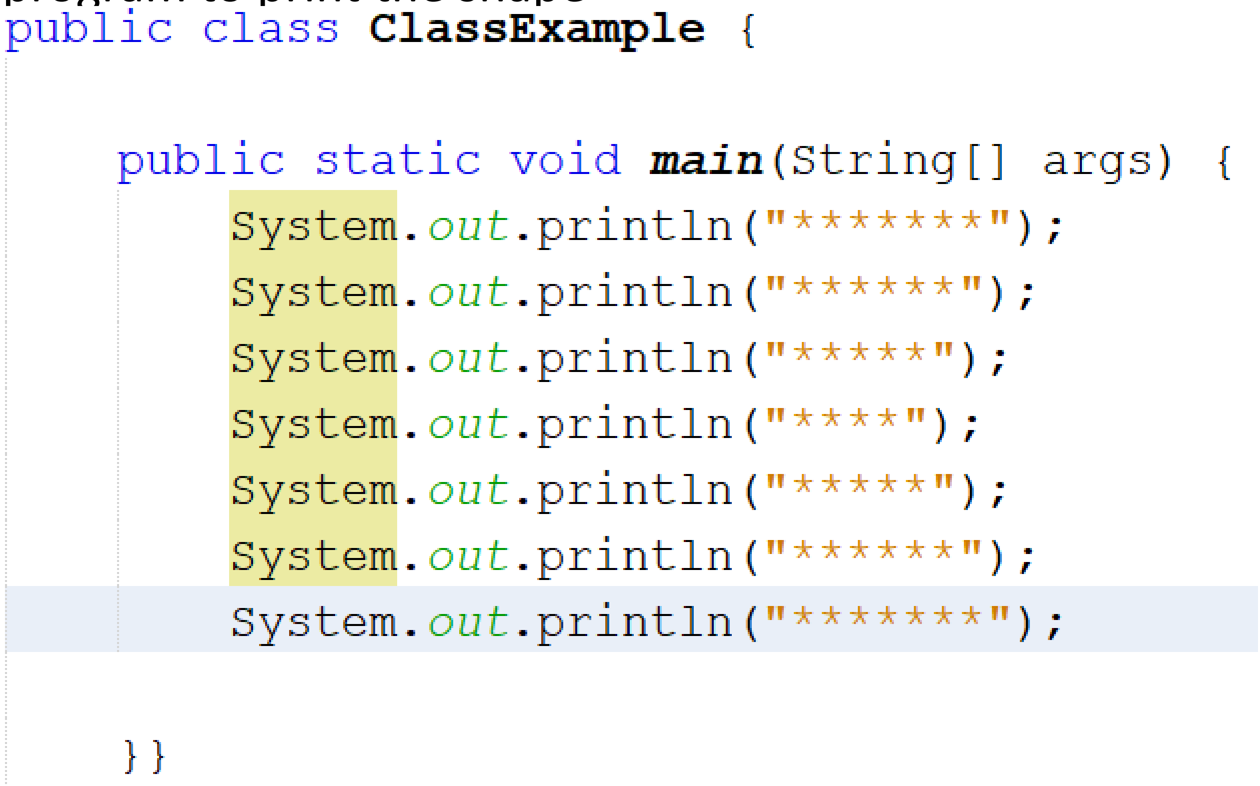
```
public class ClassExample {

    public static void main(String[] args) {
        System.out.println("*\n**\n***\n****\n*****\n******");

    }}
```



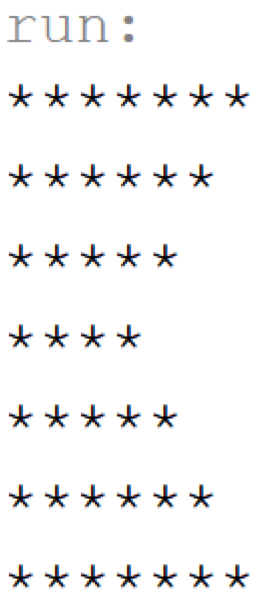
```
run:
*
**
***
****
*****
******
```

Sample Development

- Write a java program to print the shape



```
public class ClassExample {

    public static void main(String[] args) {
        System.out.println("*******");
        System.out.println("******");
        System.out.println("*****");
        System.out.println("****");
        System.out.println("*****");
        System.out.println("******");
        System.out.println("*******");

    }}
```

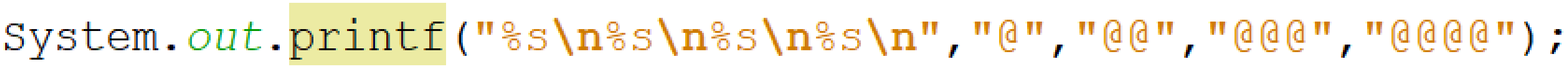


```
run:
*******
******
*****
****
*****
******
*******
```

Sample Development

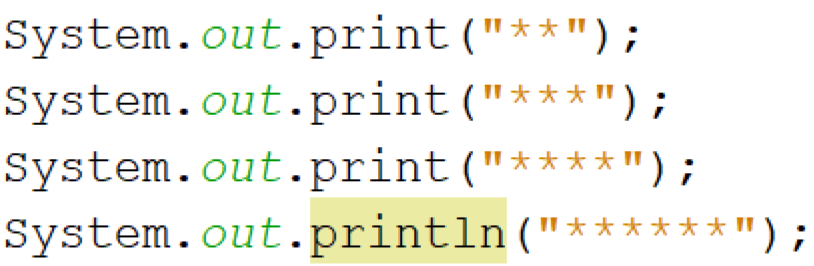
Run the following codes below in the main method and see the output.

1.



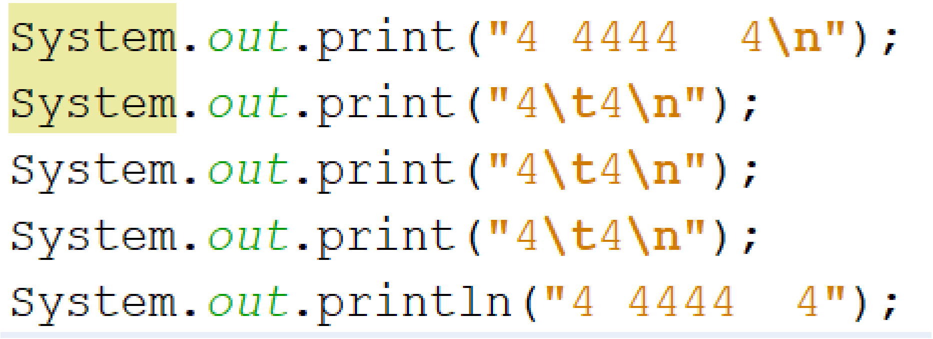
```
System.out.printf("%s\n%s\n%s\n%s\n","@","@@","@@@","@@@@");
```

2.



```
System.out.print("**");
System.out.print("***");
System.out.print("****");
System.out.println("******");
```

3.

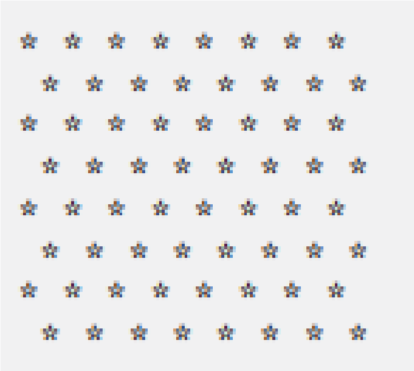


```
System.out.print("4 4444  4\n");
System.out.print("4\t4\n");
System.out.print("4\t4\n");
System.out.print("4\t4\n");
System.out.println("4 4444  4");
```

Sample Development

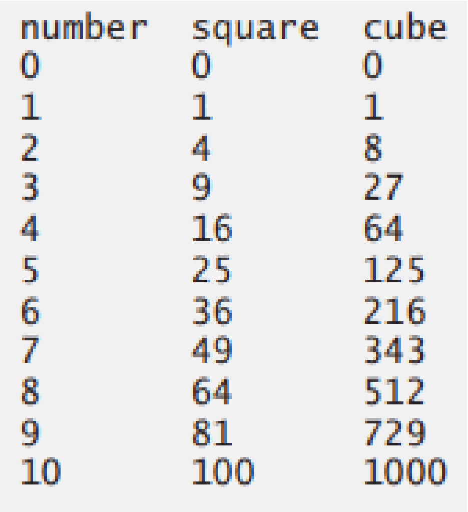
- Write java code to print the following output, try all types of print.

1\.



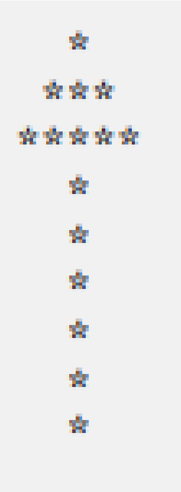
```
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
```

2\.



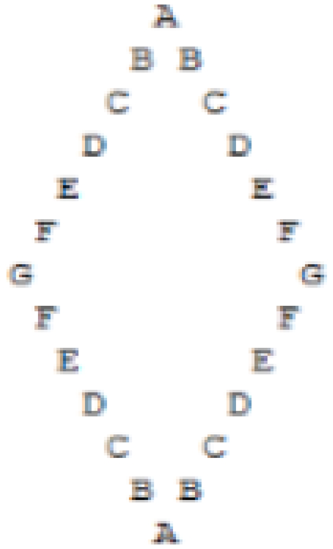
```
number  square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

3\.



```
  *
 ***
*****
  *
  *
  *
  *
  *
  *
```

4\.



```
      A
     B B
    C   C
   D     D
  E       E
 F         F
G           G
 F         F
  E       E
   D     D
    C   C
     B B
      A
```