Displaying Output

Displaying Output

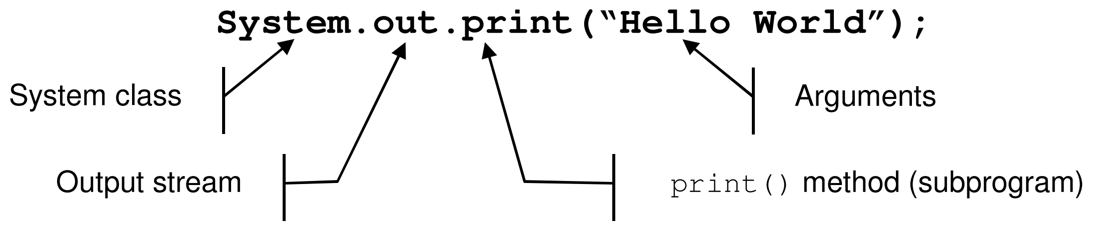
An ***output stream*** is a data channel to the operating system, which redirects the bytes of data to a specific hardware device. Java is preconfigured with several standard streams:

- **Output Stream** (`System.out`)
- **Input Stream** (`System.in`)
- **Error stream** (`System.err`)

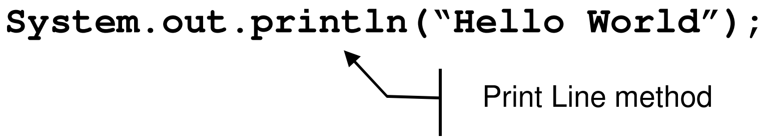
Standard output is usually handled by these Java statements

- **`System.out.print()`**
- **`System.out.println()`**
- **`System.out.format()`**

Standard Output



```
System.out.print("Hello World");
```



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

Terms

class stream method arguments

Escape sequence

- a backslash (\) followed by a symbol
- together the backslash and symbol are special output characters

`\n` ____

`\t` ____

`\\` ____

`\"` ____

Formatted Output

A new and more powerful method for text output recently became available. This is the `System.out.format` command.

`System.out.format(`*`formatString`*`, `*`data`*`);`

This command has 2 parameters: the Format String and then the data. For example:

```
String message = “Number:”;
int       n = 47;
System.out.format(“%s%d”, message, n);
```

Format String **`%[alignment][width]s`**

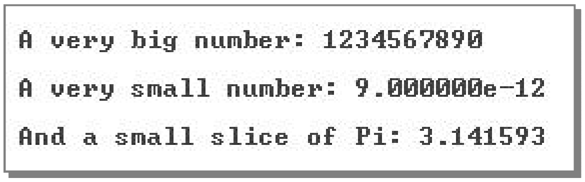
- The `format()` method's first parameter is a string that specifies (gives the details) how to convert a number.
- Any plain text in the quotations is output exactly as typed.

`%`	mark the start of the ***format specifier***
`[alignment]`	Add a minus sign (**-**) to change to left alignment or skip for right align
`[width]`	The number of characters to use for output
`s`	The result is formatted as a String
`d`	The result is formatted as a decimal integer
`f`	The result is formatted as a floating point number
`e`	formatted as a decimal number in computerized scientific notation
`t`	Prefix for date and time conversion characters (see `example4.java`)
`n`	The result is the platform-specific line separator

Example

The statements in this program segment will produce the following output:

```
long veryBig = 1234567890;
double verySmall = 0.000000000009;
System.out.format ("%n A very big number: %d %n", veryBig );
System.out.format ("%n A very small number: %e %n", verySmall );
System.out.format ("%n And a small slice of Pi: %f %n%n", Math.PI );
```



```
A very big number: 1234567890
A very small number: 9.000000e-12
And a small slice of Pi: 3.141593
```

Formatting Date and Time data

Month and day names are ***locale-specific***. This means that the names will be correct for the region as configured in the operating system.

`%`	Mark the start of the ***format specifier***
`t`	Prefix for date and time conversion characters
Date	
`B / b`	Full / abbreviated month name, e.g. "January" / "Jan"
`A / a`	Full / short name of the day of the week, e.g. "Sunday" / "Sun"
`Y / y`	Four digit / two digit year, e.g. "2009" / "09"
`j`	Day of year three digits with leading zeros as necessary, e.g. 001 - 366
`m`	Month, two digits with leading zeros as necessary, i.e. 01 - 13.
`d`	Day of month, two digits with leading zeros as necessary, i.e. 01 - 31
`e`	Day of month, formatted as two digits, i.e. 1 – 31
Time	
`H`	Hour of the day for the 24-hour clock, i.e. 00 - 23.
`I`	Hour for the 12-hour clock, i.e. 01 - 12.
`M`	Minute within the hour, i.e. 00 - 59.
`S`	Seconds within the minute, i.e. 00 – 59
`p`	Locale-specific morning or afternoon marker in lower case, e.g. "am" or "pm".

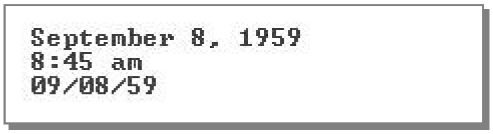
Example

```
public class Example3b {
  public static void main(String[] args)  {

          // create a Calendar object called "start"
    Calendar start = Calendar.getInstance();

        // set the date and time to Barton's first day of school
    start.set(1959,8,8,8,45,0);

    System.out.format("%n  %tB %te, %tY%n  ", start, start, start);
    System.out.format("%tl:%tM %tp%n  ", start, start, start);
    System.out.format("%tD%n%n  ", start);
  }
}
```



```
September 8, 1959
8:45 am
09/08/59
```

Activity #4

1 Use JCreator to create a new project and file called **Example4**.

2 Type this program using the indentation style shown in the previous examples

```
public class Example4{public static void main (String args[])
{System.out.print(“The unexamined life”); System.out.println(“is not
worth living\n\tSocrates”);}}
```

Activity #5

1 Use JCreator to create a new project and file called **Example5.**

2 The first 4 lines of your program must be comments with your name, course code, today’s date and the filename. Here is an example

```
// Author:            Mr. Young
// Course:            ICS 4C or 4U
// Date:              Sept. 25, 2015
// Filename:          Example5.java
```

3 Write a new Java program that will print the following exactly as shown.

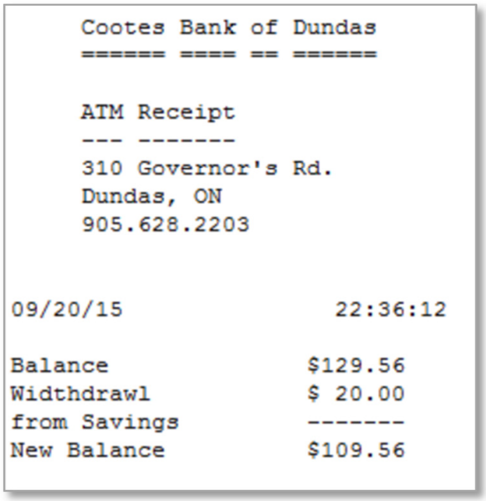
```
A slash is “/”
and
a backslash is “\”
```

Activity #6

1 Use JCreator to create a new project and file called **Example6.**

2 Add the same comments to this program as Example5.

3 Write a Java program to display an ATM receipt like this. Your receipt will have today’s date and the current time.



```
     Cootes Bank of Dundas
     ====== ==== == ======

     ATM Receipt
     --- -------
     310 Governor's Rd.
     Dundas, ON
     905.628.2203


09/20/15               22:36:12

Balance              $129.56
Widthdrawl           $ 20.00
from Savings         -------
New Balance          $109.56
```