

Course: Object Based Modeling

Code: CS-33105

Branch: MCA-3

Lecture #5 and Tutotial#1: Input and Output

Dr. J Sathish Kumar (JSK)

(Faculty & Coordinator)

Department of Computer Science and Engineering

Motilal Nehru National Institute of Technology Allahabad,

Prayagraj-211004

Reading Input

- It is easy to print output to the “standard output stream” (that is, the console window) just by calling *System.out.println*.
- Reading from the “standard input stream” *System.in* isn’t quite as simple.
- To read console input, you first construct a Scanner that is attached to *System.in*:
 - `Scanner in = new Scanner(System.in) ;`
- Learn to use various methods of the Scanner class to read input

nextLine method

- For example, the nextLine method reads a line of input.

```
System.out.print("What is your name? ");
```

```
String name = in.nextLine();
```

- Here, we use the nextLine method because the input might contain spaces.
- To read a single word (delimited by whitespace), call

```
String firstName = in.next();
```

- To read an integer, use the nextInt method.

```
System.out.print("How old are you? ");
```

```
int age = in.nextInt();
```

- Similarly, the nextDouble method reads the next floating-point number
- `import java.util.*;`

Example#1: InputTest/InputTest.java

```
1  import java.util.*;
2
3  /**
4   * This program demonstrates console input.
5   * @version 1.10 2004-02-10
6   * @author Cay Horstmann
7   */
8  public class InputTest
9  {
10     public static void main(String[] args)
11     {
12         Scanner in = new Scanner(System.in);
13
14         // get first input
15         System.out.print("What is your name? ");
16         String name = in.nextLine();
17
18         // get second input
19         System.out.print("How old are you? ");
20         int age = in.nextInt();
21
22         // display output on console
23         System.out.println("Hello, " + name + ". Next year, you'll be
24     }
25 }
```

java.util.Scanner

- `Scanner(InputStream in)`
 - constructs a Scanner object from the given input stream.
- `String nextLine()`
 - reads the next line of input.
- `String next()`
 - reads the next word of input (delimited by whitespace).
- `int nextInt()`
- `double nextDouble()`
 - reads and converts the next character sequence that represents an integer or floating-point number.
- `boolean hasNext()`
 - tests whether there is another word in the input.
- `boolean hasNextInt()`
- `boolean hasNextDouble()`
 - tests whether the next character sequence represents an integer or floating point number.

Java Programming Exercises #1

- Write a Java program to compute the area of a hexagon.
 - Area of a hexagon = $(6 * s^2) / (4 * \tan(\pi/6))$
where s is the length of a side
 - Input Data:
Input the length of a side of the hexagon: 6
 - Expected Output
The area of the hexagon is: 93.53074360871938

Java Programming Exercises Solution#1

```
1.  import java.util.Scanner;
2.  public class Exercise1
3.  {
4.      public static void main(String[] args)
5.      {
6.          Scanner input = new Scanner(System.in);
7.          System.out.print("Input the length of a side of the hexagon: ");
8.          double s = input.nextDouble();
9.          System.out.print("The area of the hexagon is: " + hexagonArea(s)+"\n");
10.     }
11.     public static double hexagonArea(double s)
12.     {
13.         return (6*(s*s))/(4*Math.tan(Math.PI/6));
14.     }
15. }
```

Formatting Output

- You can print a number *x* to the console with the statement *System.out.print(x)* .
- That command will print *x* with the maximum number of nonzero digits for that type.

- For example,
double x = 10000.0 / 3.0;
System.out.print(x) ;
prints
3333.333333333335

- In early versions of Java, formatting numbers was a bit of a hassle. Fortunately, Java 5 brought back the venerable `printf` method from the C library.

- For example, the call

System.out.printf("%8.2f", x) ;

prints *x* with a *field width* of 8 characters and a *precision* of 2 characters. That is, the printout contains a leading space and the seven characters
3333.33

- You can supply multiple parameters to `printf`. For example

System.out.printf("Hello, %s. Next year, you' ll be %d", name, age) ;

Conversion Character	Type	Example
d	Decimal integer	159
x	Hexadecimal integer	9f
o	Octal integer	237
f	Fixed-point floating-point	15.9
e	Exponential floating-point	1.59e+01
g	General floating-point (the shorter of e and f)	—
a	Hexadecimal floating-point	0x1.fccdp3
s	String	Hello
c	Character	H
b	boolean	true
h	Hash code	42628b2
t or T	Date and time (T forces uppercase)	Obsolete, use the <code>java.time</code> classes instead—see Chapter 6 of Volume II
%	The percent symbol	%
n	The platform-dependent line separator	—

Conversions for printf

Flags for printf

Flag	Purpose	Example
+	Prints sign for positive and negative numbers.	+3333.33
space	Adds a space before positive numbers.	 3333.33
0	Adds leading zeroes.	003333.33
-	Left-justifies field.	3333.33
(Encloses negative numbers in parentheses.	(3333.33)
,	Adds group separators.	3,333.33
# (for f format)	Always includes a decimal point.	3,333.
# (for x or o format)	Adds 0x or 0 prefix.	0xcafe
\$	Specifies the index of the argument to be formatted; for example, %1\$d %1\$x prints the first argument in decimal and hexadecimal.	159 9F
<	Formats the same value as the previous specification; for example, %d %<x prints the same number in decimal and hexadecimal.	159 9F

Java Programming Exercises #2

- Write a Java program to calculate the sum of two integers and return true if the sum is equal to a third integer.

Sample Output:

Input the first number : 5

Input the second number: 10

Input the third number : 15

The result is: true

```
1.  import java.util.*;
2.  public class Exercise2 {
3.  public static void main(String[] args)
4.  {
5.      Scanner in = new Scanner(System.in);
6.      System.out.print("Input the first number : ");
7.      int x = in.nextInt();
8.          System.out.print("Input the second number: ");
9.          int y = in.nextInt();
10.         System.out.print("Input the third number : ");
11.         int z = in.nextInt();
12.         System.out.print("The result is: "+sumoftwo(x, y, z));
13.         System.out.print("\n");
14.     }
15.
16.     public static boolean sumoftwo(int p, int q, int r)
17.     {
18.         return ((p + q) == r || (q + r) == p || (r + p) == q);
19.     }
20. }
```

Java Programming Exercises Solution#2

Tutorial #1

1. Write a Java program to compute the sum of the first 100 prime numbers. (Sum of the first 100 prime numbers: 24133)
2. Write a Java program to test if the first and the last element of two array of integers are same. The length of the array must be greater than or equal to 2.

(Test Data: array1 = 50, -20, 0, 30, 40, 60, 12

array2 = 45, 20, 10, 20, 30, 50, 11

Sample Output:false)

3. Write a Java program to merge two given sorted array of integers and create a new sorted array.

array1 = [1,2,3,4]

array2 = [2,5,7, 8]

result = [1,2,2,3,4,5,7,8]