

User I/O

For any program to function appropriately and make it interactive, input and output is the most. Input is taken from the user and based on which the output is shown to the user. These inputs and outputs are handled by some specific functions, or classes or attributes which might differ from programming language to language. Here, in JAVA for performing the User I/O there are specific function and class which are used when required.

User Output

Now talking about the output in JAVA, we have already studied about some of the statements in the previous lecture as well. Along with this, there are others which are as;

```
System.out.println(); or  
System.out.print(); or  
System.out.printf();
```

Here, System is a class, out is a public static field which accepts the output data and the last keyword used in each statement are the functions which print the output as required.

Let us take an example as;

```
class Demo {  
    public static void main(String[] args) {  
        System.out.println("Java programming is easy.");  
    }  
}
```

Output

```
Java programming is easy.
```

In the above example we used println() method or function. Beside this there are two more functions which prints output differently.

1. print() - It prints string inside the quotes.
2. println() - It prints string inside the quotes similar like print() method. Then the cursor moves to the beginning of the next line.
3. printf() - It provides string formatting (similar to printf in C/C++ programming).

```
class Output {  
    public static void main(String[] args) {  
  
        System.out.println("1. println ");  
        System.out.println("2. println ");  
  
        System.out.print("1. print ");  
        System.out.print("2. print");  
    }  
}
```

Output

```
1. println
2. println
1. print 2. print
```

Note: Research on your own for knowing about the `printf()` method.

User Input

Similarly, for taking the user input, there are various ways in JAVA. Among all them we will learn about **Scanner** class.

The **Scanner** class is used to get user input, and it is found in the **java.util package**.

To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class. These methods might be different when we take input of different data types.

For example, we will use the `nextLine()` method, which is used to read Strings, `nextInt()` method for taking int value as input and so on.

Method	Description
<code>nextBoolean()</code>	Reads a boolean value from the user
<code>nextByte()</code>	Reads a byte value from the user
<code>nextDouble()</code>	Reads a double value from the user
<code>nextFloat()</code>	Reads a float value from the user
<code>nextInt()</code>	Reads a int value from the user
<code>nextLine()</code>	Reads a String value from the user
<code>nextLong()</code>	Reads a long value from the user
<code>nextShort()</code>	Reads a short value from the user


Now let us see an example as;

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter name, age and salary:");
        // String input
        String name = myObj.nextLine();
        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();
        // Output input by user
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
    }
}
```

Output

```
Enter name, age and salary:
ujjwal
28
1000000
Name: ujjwal
Age: 28
Salary: 1000000.0
```



Task to Do:

- 1. Write a program to take student details as input and display the result.**
- 2. Write a program to calculate sum of four numbers taking user input.**
- 3. Write a program to take input of the total marks of four subjects of a student and calculate the total percentage secured. Then display the percentage and final result of the student;**
 - a. If equal to or more than 70 -> First Class**
 - b. If more than 59 -> Upper second Class**
 - c. If more than 49 -> Second class**
 - d. If more than 39 -> Third class and if below than 40 the result is fail.**
- 4. Write a program to take two integer inputs from user and print sum and product of them.**
- 5. Take two integer inputs from user. First calculate the sum of two and then product. Finally, calculate the division of thus obtained sum and product and print the result.**
- 6. Ask user to give two double input for length and breadth of a rectangle and print area type casted to int.**
- 7. Take name, roll number and field of interest from user and print in the format below :Hey, my name is xyz and my roll number is xyz. My field of interest are xyz.**
- 8. Take side of a square from user and print area and perimeter of it. Also, calculate SI, Area of triangle and Volume of Cube and Cuboid. Take the attributes as user input.**
- 9. Write a program to find square of a number.**

E.g.- INPUT : 2 OUTPUT : 4

INPUT : 5 OUTPUT : 25

- 10. Take two different string input and print them in same line. E.g.-**

INPUT : Codes

Dope

OUTPUT : CodesDope

- 11. Take 3 inputs from user and check :**

all are equal

any of two are equal

(use && || with ternary operator)

- 12. Write a program to enter the values of two variables 'a' and 'b' from keyboard and then check if both the conditions 'a < 50' and 'a < b' are true.**

- 13. If the marks of Robert in three subjects are entered through keyboard (each out of 100), write a program to calculate his total marks and percentage marks.**

Parsing and Parse methods in JAVA

Parsing in its most general sense is the extraction of the necessary information from some piece of data, most often textual data.

There are many Java classes that have the `parse()` method. Usually the `parse()` method receives some string as input, "extracts" the necessary information from it and converts it into an object of the calling class.

For example, it received a string and returned the date that was "hiding" in this string.

Here, we will learn about some useful variations of `parse()`.

Let's start from one of the most popular `parse()` methods, that is not exactly `parse()`, but `parseInt()`.

Java `parseInt()` method is used to get the primitive data type from a specific string. In other words it converts a string to a number. `parseInt()` can have one or two arguments. Here's the syntax of `parseInt()`:

```
static int parseInt(String s)
static int parseInt(String s, int radix)
```

Where `s` is the string representing a signed decimal value and `radix` the base of a numerical system.

Keep in mind that there's no default base value — you need to enter one within the range of 2 and 36. Here's an example. How to parse with `parseInt()`:

```
public class ParseInt {

    public static void main(String args[]){
        int x = Integer.parseInt("12");
        double c = Double.parseDouble("12");
        int b = Integer.parseInt("100",2);

        System.out.println(Integer.parseInt("12"));
        System.out.println(Double.parseDouble("12"));
        System.out.println(Integer.parseInt("100",2));
        System.out.println(Integer.parseInt("101", 8));

    }
}
```

The output is:

12 12.0 4 65

Examples of other methods are;

// Store 1 in bVar.

```
byte bVar = Byte.parseByte("1");
```

// Store 2599 in iVar.

```
int iVar = Integer.parseInt("2599");
```

// Store 10 in sVar.

```
short sVar = Short.parseShort("10");
```

// Store 15908 in lVar.

```
long lVar = Long.parseLong("15908");
```

// Store 12.3 in fVar.

```
float fVar = Float.parseFloat("12.3");
```

// Store 7945.6 in dVar.

```
double dVar = Double.parseDouble("7945.6");
```

Use of Dialog Boxes for User I/O

A dialog box is a small graphical window that displays a message to the user or requests input.

A variety of dialog boxes can be displayed using the **JOptionPane** class.

Two of the dialog boxes are:

- Message Dialog - a dialog box that displays a message.
- Input Dialog - a dialog box that prompts the user for input.

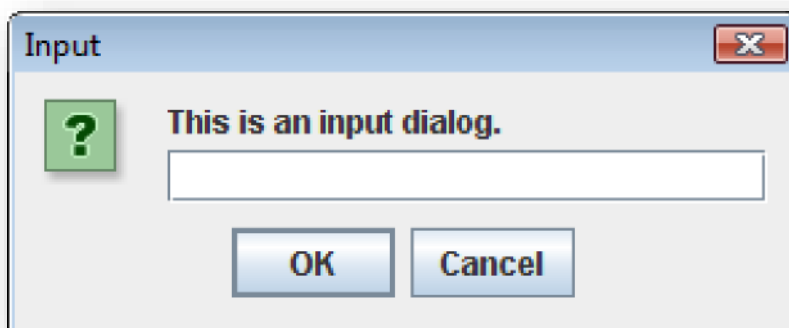
JOptionPane Class

The JOptionPane class is not automatically available to your Java programs. The following statement must be declared or imported before the program's class header:

import javax.swing.JOptionPane;

This statement tells the compiler where to find the JOptionPane class and use it in your program. The JOptionPane class provides methods to display each type of dialog box.

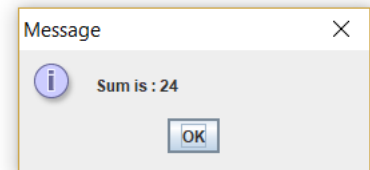
For example;



Example of Message Dialog:

```
import javax.swing.JOptionPane;
import java.util.Scanner;

class AddTwoNos
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int first , second , sum;
        System.out.println("Enter first number : ");
        first = sc.nextInt();
        System.out.println("Enter second number : ");
        second = sc.nextInt();
        sum = first + second;
        JOptionPane.showMessageDialog(null, "Sum is : " + sum);
    }
}
```



Example of Input Dialog:

```
import javax.swing.JOptionPane;
class AddTwoNos
{
    public static void main(String args[])
    {

        int first , second , sum;

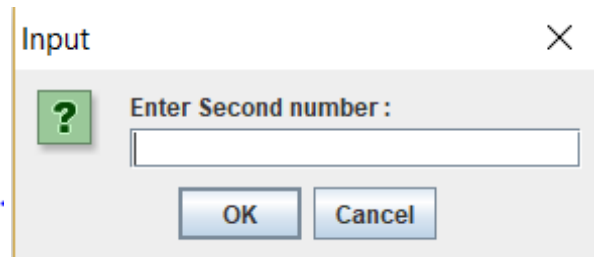
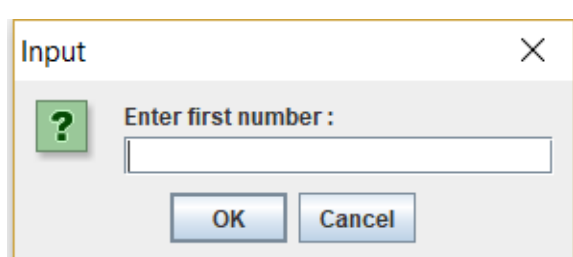
        first = Integer.parseInt(JOptionPane.showInputDialog( "Enter first number : " ));
        second = Integer.parseInt(JOptionPane.showInputDialog( "Enter Second number : " ));

        sum = first + second;

        JOptionPane.showMessageDialog(null, "Sum is : " + sum);

    }
}
```

Output



System.exit Method

A program that uses **JOptionPane** does not automatically stop executing when the end of the main method is reached. Java generates a *thread*, which is a process running in the computer, when a **JOptionPane** is created. If the **System.exit** method is not called, this thread continues to execute. So, in order to stop this from happening we use **System.exit** method.

Some important things to consider are;

1. The **System.exit** method requires an integer argument.

System.exit(0);

2. This argument is an *exit code* that is passed back to the operating system.
3. This code is usually ignored; however, it can be used outside the program: to indicate whether the program ended successfully or as the result of a failure.
4. The value **0** traditionally indicates that the program ended successfully.

Example showing the use of **System.exit** method;

```
import javax.swing.*;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter name, age and salary:");
        // String input
        String name = myObj.nextLine();
        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();
        // Output input by user
        JOptionPane.showMessageDialog(null, "Name: "+name+"\nAge: "+age+"\nSalary: "+salary);
        System.exit(0);
    }
}
```

Output

Enter name, age and salary:

ujjwal

29

100000



Task to Do: *Use JOptionPane for User I/O*

1. Write a program to take student details as input and display the result.
2. Write a program to calculate sum of four numbers taking user input.
3. Write a program to take input of the total marks of four subjects of a student and calculate the total percentage secured. Then display the percentage and final result of the student;
If equal to or more than 70 -> First Class
If more than 59 -> Upper second Class
If more than 49 -> Second class
If more than 39 -> Third class and if below than 40 the result is fail.
4. Write a program to take two integer inputs from user and print sum and product of them.
14. Take two integer inputs from user. First calculate the sum of two then product of two.
Finally, calculate the division of thus obtained sum and product and print the result.
5. Ask user to give two double input for length and breadth of a rectangle and print area type casted to int.
6. Take name, roll number and field of interest from user and print in the format below :Hey, my name is xyz and my roll number is xyz. My field of interest are xyz.
7. Take side of a square from user and print area and perimeter of it. Also, calculate SI, Area of triangle and Volume of Cube and Cuboid. Take the attributes as user input.
8. Write a program to find square of a number.
E.g.- INPUT : 2 OUTPUT : 4
INPUT : 5 OUTPUT : 25
9. Take two different string input and print them in same line. E.g.-
INPUT : Codes
Dope
OUTPUT : CodesDope
10. Take 3 inputs from user and check :
all are equal
any of two are equal
(use && || with ternary operator)
11. Write a program to enter the values of two variables 'a' and 'b' from keyboard and then check if both the conditions 'a < 50' and 'a < b' are true.
12. If the marks of Robert in three subjects are entered through keyboard (each out of 100), write a program to calculate his total marks and percentage marks.