

How to Read and Print an Integer value in Java

- The given task is to take an integer as input from the user and print that integer in Java language.

Read and Print an Integer value in Java

```
int varName  
↓  
Scanner se = new Scanner()  
↓  
varName = se.nextInt()  
↓  
System.out.print(varName)
```



In the below program, the syntax and procedures to take the integer as input from the user are shown in Java language.

Steps for Input

1. The user enters an integer value when asked.
2. This value is taken from the user with the help of **nextInt()** method of **Scanner Class**.
The nextInt() method, in Java, reads the next integer value from the console into the specified variable.

Syntax of Scanner Class

```
variableOfIntType = ScannerObject.nextInt();
```

where **variableOfIntType** is the variable in which the input value is to be stored.
And **ScannerObject** is the beforehand created object of the Scanner class.

Steps for Output

1. This entered value is now stored in the **variableOfIntType**.
2. Now to print this value, **System.out.println()** or **System.out.print()** method is used.
The System.out.println() method, in Java, prints the value passed as the parameter to it, on the console screen and the changes the cursor to the next line on the console.
Whereas System.out.print() method, in Java, prints the value passed as the parameter to

it, on the console screen and the cursor remains on the next character of the last printed character on the console.

Syntax of println

```
System.out.println(variableOfType);
```

Hence, the integer value is successfully read and printed.

Program Read and Print an Integer value in Java

Below is the implementation of the above method:

- Java

```
// Java program to take an integer

// as input and print it

import java.io.*;

import java.util.Scanner;

// Driver Class

class Accept {

    // main function

    public static void main(String[] args)

    {

        // Declare the variables

        int num;
```

```
// Input the integer

System.out.println("Enter the integer: ");

// Create Scanner object

Scanner s = new Scanner(System.in);

// Read the next integer from the screen

num = s.nextInt();

// Display the integer

System.out.println("Entered integer is: " + num);

}

}
```

Input

Enter the integer: 10

Output

Entered integer is: 10

2. // Java Program to print Multiplication of two floating
// point Number.

```
import java.io.*;
```

```
class Multiply {  
    public static void main(String[] args)  
    {  
  
        // f is to ensures that numbers are float DATA TYPE  
        float f1 = 1.5f;  
        float f2 = 2.0f;  
  
        // to store the multiplied value  
        float p = f1 * f2;  
  
        // to print the product  
        System.out.println("The product is: " + p);  
    }  
}
```

Output

```
The product is: 3.0
```

3.

```
// Java Program to Swap Two values using third variable  
// using temp variable
```

```
// Importing generic libraries
import java.util.*;

class Swap {
    // Function to swap two numbers
    // Using temporary variable
    static void swapValuesUsingThirdVariable(int m, int n)
    {
        // Swapping the values
        int temp = m;
        m = n;
        n = temp;
        System.out.println("Value of m is " + m + " and Value of n is " + n);
    }

    // Main driver code
    public static void main(String[] args)
    {
        // Random integer values
        int m = 9, n = 5;

        // Calling above function to
        // reverse the numbers
        swapValuesUsingThirdVariable(m, n);
    }
}
```

```
}
```

output

```
Value of m is 5 and Value of n is 9
```

4.

```
// Java Program to Check if Given Integer is Odd or Even
```



```
// Importing required classes
```

```
import java.io.*;
```

```
import java.util.Scanner;
```



```
// Main class
```

```
class Odd {
```



```
    // Main Driver Method
```

```
    public static void main(String[] args)
```

```
    {
```

```
        // Declaring and initializing integer variable
```

```
        int num = 10;
```



```
        // Checking if number is even or odd number
```

```
        // via remainder
```

```
        if (num % 2 == 0) {
```



```
            // If remainder is zero then this number is even
```

```
        System.out.println("Entered Number is Even");  
    }  
  
    else {  
  
        // If remainder is not zero then this number is  
        // odd  
        System.out.println("Entered Number is Odd");  
    }  
}  
}
```

Output

```
Entered Number is Even
```

5.

```
// Java program to find all the  
// prime numbers from 1 to N  
class Prime {  
  
    // Function to print all the  
    // prime numbers till N  
    static void prime_N(int N)  
    {  
        // Declaring the variables  
        int x, y, flg;
```

```
// Printing display message
System.out.println("All the Prime numbers within 1 and " + N +
"are:");
// Using for loop for traversing all
// the numbers from 1 to N
for (x = 1; x <= N; x++) {

    // Omit 0 and 1 as they are
    // neither prime nor composite
    if (x == 1 || x == 0)
        continue;

    // Using flag variable to check
    // if x is prime or not
    flg = 1;

    for (y = 2; y <= x / 2; ++y) {
        if (x % y == 0) {
            flg = 0;
            break;
        }
    }

    // If flag is 1 then x is prime but
    // if flag is 0 then x is not prime
```

```

        if (flg == 1)
            System.out.print(x + " ");
    }

}

// The Driver code
public static void main(String[] args)
{
    int N = 45;

    prime_N(N);
}
}

```

Output

All the Prime numbers within 1 and 45 are:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43
```

6.

```

// JAVA program to find Armstrong
// numbers between two integers
import java.io.*;
import java.math.*;

class Armstrong {

```

```
// Function to print Armstrong
// Numbers between two integers
static void ArmstrongNum(int l, int h)
{
    for (int j = l + 1; j < h; ++j) {

        // Calculating number of digits
        int y = j;
        int N = 0;
        while (y != 0) {
            y /= 10;
            ++N;
        }

        // Calculating the sum of nth
        // power of all the digits
        int sum_power = 0;
        y = j;
        while (y != 0) {
            int d = y % 10;
            sum_power += Math.pow(d, N);
            y /= 10;
        }

        // Checking if the current number
```

```

        // i is equal to the sum of nth
        // power of all the digits
        if (sum_power == j)
            System.out.print(j + " ");
    }

}

// The Driver code
public static void main(String args[])
{
    int n1 = 50;
    int n2 = 500;
    ArmstrongNum(n1, n2);
    System.out.println();
}

```

Output

```
153 370 371 407
```

7.

```

// java program to check whether input
// character is a vowel or consonant

import java.io.*;

public class Vowel {

```

```

// Function to find whether an input
// character is vowel or not
static void Vowel_Or_Consonant(char y)
{
    if (y == 'a' || y == 'e' || y == 'i' || y == 'o'
        || y == 'u')
        System.out.println("It is a Vowel.");
    else
        System.out.println("It is a Consonant.");
}

// The Driver code
static public void main(String[] args)
{
    Vowel_Or_Consonant('b');
    Vowel_Or_Consonant('u');
}

```

Output

It is a Consonant.

It is a Vowel.

8.

```

// Java program to find compound interest for
// given values.

```

```
import java.io.*;  
  
class CompoundInterest  
{  
    public static void main(String args[])  
    {  
        double principal = 10000, rate = 10.25, time = 5;  
  
        /* Calculate compound interest */  
        double CI = principal *  
                    (Math.pow((1 + rate / 100), time));  
  
        System.out.println("Compound Interest is "+ CI);  
    }  
}  
// This code is contributed by Anant Agarwal.
```

Output

```
Compound Interest is 16288.946267774416
```

9.

```
// Java program to find the perimeter of a Rectangle
```

```
import java.io.*;
```

```
class Perimeter {  
  
    // Method to calculate the perimeter of the rectangle  
    // with given length and breadth  
    static void perimeter(int length, int breadth)  
    {  
        // Calculate the 'perimeter' using the formula  
        int perimeter = 2 * (length + breadth);  
  
        System.out.println("The perimeter of the given  
rectangle of length "  
            + length + " and breadth " + breadth + " = "  
            + perimeter);  
    }  
  
    // Driver method  
    public static void main(String[] args)  
    {  
        // Initialize a variable length that stores length of  
        // the given rectangle  
        int length = 10;  
  
        // Initialize a variable breadth that stores breadth  
        // of the given rectangle  
        int breadth = 20;
```

```
        // Call the perimeter method on these length and  
        // breadth  
        perimeter(length, breadth);  
    }  
}
```

Output

```
The perimeter of the given rectangle of length 10 and breadth 20 = 60
```

10

Right Triangle Star Pattern:

```
Input : n = 5
```

```
Output:
```

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

```
import java.io.*;
```

```
// Java code to demonstrate right star triangle  
public class Pattern {  
    // Function to demonstrate printing pattern  
    public static void StarRightTriangle(int n)  
    {  
        int a, b;  
  
        // outer loop to handle number of rows
```

```
// k in this case
for (a = 0; a < n; a++) {

    // inner loop to handle number of columns
    // values changing acc. to outer loop
    for (b = 0; b <= a; b++) {
        // printing stars
        System.out.print("* ");

    }

    // end-line
    System.out.println();
}

}

// Driver Function
public static void main(String args[])
{
    int k = 5;
    StarRightTriangle(k);
}

}
```

Output

```
*
```

```
* *
```

```
* * *
```

* * * *

* * * * *

11. Programs to find the numbers and sum of them which is > 100 and < 200 that are divisible by 7.

```
public class Dseven
{
    public static void main (String [] args)
    {
        int lowerlimit = 100; int upperlimit = 200;
        int divisibleby = 7; int count = 0; int sum = 0;
        for (int i = lowerlimit; i < upperlimit; i++)
        {
            if (i % divisibleby == 0)
            {
                count++;
                sum += i;
            }
        }
        System.out.println ("Numbers " + count);
        System.out.println ("Sum = " + sum);
    }
}
```

12. Program to check a num is palindrome using command line argument

```
public class Palindrome
{
    public static void main (String [] args)
    {
        if (args.length != 1)
        {
            System.out.println ("Usage : Java
                Palindrom Check <number> ");
            return;
        }

        String inputString = args[0];
        int number = Integer.parseInt (inputString);
        String original = Integer.toString (number);
        String reversed = new StringBuilder (original)
            .reverse ().toString ();

        if (original.equals (reversed))
        {
            System.out.println (number + " is a
                palindrome ");
        }
        else
        {
            System.out.println (number + " is not
                palindrome ");
        }
    }
}
```

Ques. Program to perform constructor overloading

```
public class ConstructorOverloading
{
    private int value;
    public ConstructorOverloading()
    {
        this.value = 0;
    }
    public ConstructorOverloading(int newValue)
    {
        this.value = newValue;
    }
    public int getValue()
    {
        return value;
    }
}
public static void main(String[] args)
{
    ConstructorOverloading obj1 = new
        ConstructorOverloading();
    ConstructorOverloading obj2 = new
        ConstructorOverloading();
    System.out.println("Value obj1: " +
        obj1.getValue());
    System.out.println("Value obj2: " + obj2.getValue());
}
```

14. Program to calculate the volume of different objects using method overloading.

```
public class VolumeCalculation
{
    public static double calculateCubeVolume(double side)
    {
        return Math.pow(side, 3);
    }

    public static double calculateCylinderVolume(double radius, double height)
    {
        return Math.PI * Math.pow(radius, 2) * height;
    }

    public static double calculateSphereVolume(double radius)
    {
        return (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
    }
}
```

```
public static void main (String [] args)
```

```
{
```

```
    System.out.println ("Volume of cube: " +  
        calculateCubeVolume (5.0));
```

```
    System.out.println ("Volume of cylinder: " +  
        calculateCylinderVolume  
        (3.0, 8.0));
```

```
    System.out.println ("Volume of sphere: " +  
        calculateSphereVolume  
        (4.0));
```

```
}
```

```
}
```

8/8

15. Program to find the area of a circle

```
class Circle
```

```
{
```

```
    public static final pi = 3.14;
```

```
    void getData (int x)
```

```
{
```

```
    int radius = x;
```

```
}
```

```
    double getArea()
```

```
{
```

```
    double area = pi * radius * radius;
```

```
    return (area);
```

```
}
```

```
}
```

```
    public static void main (String [] args)
```

```
{
```

```
    Circle cir = new Circle();
```

```
    cir.radius = 10;
```

```
    double area = circle.getArea();
```

System.out.println ("The area of circle is : "
+ area);

```
}
```

```
}
```

16. Program to call the method `display()` of class `Former` first and then call `display()` method of class `Latter`.

class Former

{

 public void display()

{

 System.out.println("This is a Former class");

}

}

class Latter

{

 public void display()

{

 System.out.println("This is Latter class");

}

}

public class Method

{

 public static void main (String args [])

{

 Former f = new Former();

 f.display();

 Latter l = new Latter();

l. display;

{

{