**Java Program to Get Input from User**

This article is created to cover multiple programs in Java that are based on receiving inputs from user. Here are the list of programs included in this article:

* Get integer input in Java
* Continue receiving inputs until user enters 0
* How to handle with invalid inputs in Java ?
* Get character input in Java
* Get string input in Java

**Get Integer Input in Java**

The question is, *write a Java program to ask the user to enter an integer value and print the entered value back on the output screen.* The program given below is its answer. This program basically shows, how to read an integer value in Java using **Scanner** and **nextInt()**

**import java.util.Scanner;**

**class Example1**

**{**

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

**{**

**int num;**

**Scanner scan = new Scanner(System.in);**

**System.out.print("Enter an Integer Value: ");**

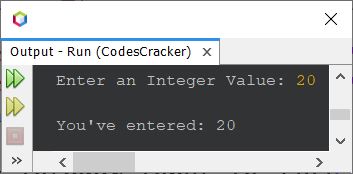
**num = scan.nextInt();**

**System.out.println("\nYou've entered: " +num);**

**}**

**}**

The snapshot given below shows the sample run of above program, with user input **20**:



You can use following methods, to scan values of other types:

* **nextDouble()** - to read value of **double** data type
* **nextFloat()** - to read value of **float** type
* **nextLong()** - to read value of **long** type
* **nextShort()** - to read value of **short** type
* **nextByte()** - to read value of **byte** type

## Swap Two Numbers using Third Variable in Java

The question is, *write a Java program to swap any two given numbers. The number must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

class Codes

{

public static void main(String[] args)

{

int a, b, temp;

Scanner s = new Scanner(System.in);

System.out.print("Enter the First Number: ");

a = s.nextInt();

System.out.print("Enter the Second Number: ");

b = s.nextInt();

temp = a;

a = b;

b = temp;

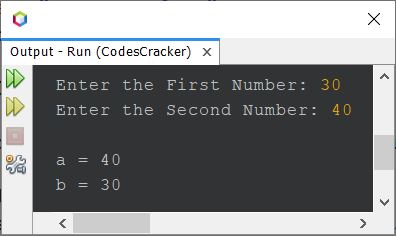
System.out.println("\na = " +a);

System.out.println("b = " +b);

}

}

The snapshot given below shows the sample run of above Java program, on swapping of two given numbers, with user input **30** as first and **40** as second number:



That is, the first number say **30** gets stored in **a** variable, and the second number say **40** gets stored in **b** variable. And using the statement:

temp = a;

Now **temp** holds the value of **a**, that is **30**. Again using the following statement:

a = b;

The value of **a** becomes **40**. And finally using the statement given below:

b = temp;

The value of **temp**, that is **30** gets initialized to **b**. So now, **b** holds the value of **a**, and **a** holds the value of **b**. That's it.

The above program can also be created in this way:

import java.util.Scanner;

class swap

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter the First Number: ");

int a = s.nextInt();

System.out.print("Enter the Second Number: ");

int b = s.nextInt();

System.out.println("\n----Before Swap----");

System.out.println("a = " +a);

System.out.println("b = " +b);

int temp = a;

a = b;

b = temp;

System.out.println("\n----After Swap----");

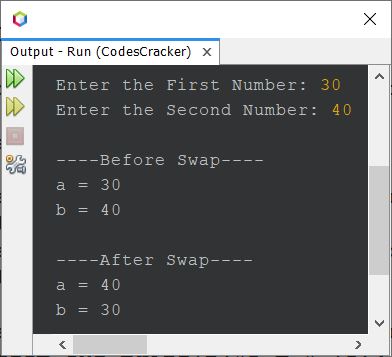
System.out.println("a = " +a);

System.out.println("b = " +b);

}

}

Here is its sample run with same user input as of previous program's sample run:



## Swap Two Numbers without using Third Variable in Java

This program does not uses any third variable like **temp** to swap two numbers. Rather it uses the simple addition and subtraction operation to do the job.

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter the First Number: ");

int numOne = s.nextInt();

System.out.print("Enter the Second Number: ");

int numTwo = s.nextInt();

System.out.println("\n----Before Swap----");

System.out.println("numOne = " +numOne);

System.out.println("numTwo = " +numTwo);

numOne = numOne + numTwo;

numTwo = numOne - numTwo;

numOne = numOne - numTwo;

System.out.println("\n----After Swap----");

System.out.println("numOne = " +numOne);

System.out.println("numTwo = " +numTwo);

}

}

## Print ASCII Value of a Given Character in Java

**Note -**ASCII values of **A-Z** are **65-90**.

**Note -**ASCII values of **a-z** are **97-122**.

**Note -**ASCII values of **0-9** are **48-57**.

The question is, *write a Java program to find and print the ASCII value of a character. The character must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

char ch;

int ascii;

Scanner scan = new Scanner(System.in);

System.out.print("Enter a Character: ");

ch = scan.next().charAt(0);

ascii = ch;

System.out.println("\nASCII Value = " +ascii);

}

}

## Add Two Numbers in Java using Scanner

The question is, *write a Java program to add any two numbers. Both the number must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

class Example2

{

public static void main(String[] args)

{

int numberOne, numberTwo, add;

Scanner s = new Scanner(System.in);

System.out.print("Enter the First Number: ");

numberOne = s.nextInt();

System.out.print("Enter the Second Number: ");

numberTwo = s.nextInt();

add = numberOne + numberTwo;

System.out.println("\nResult = " +add);

}

}

# Java Program to Calculate Area and Perimeter of a Rectangle

This article covers a program in Java that find and prints area and perimeter of a rectangle based on its size provided by user at run-time of the program.

**Note -**The area of a rectangle is calculated using the formula **len\*bre**. Where **len** and **bre** are the values of length and breadth, the two adjacent sides of rectangle, where length is greater than breadth.

**Note -**The perimeter of a rectangle is calculated using the formula **(2\*len) + (2\*bre)** or **2\*(len+bre)**.

## Find Area of Rectangle in Java

The question is, *write a program in Java that calculates area of rectangle.* The program given below is its answer:

import java.util.Scanner;

class CodesCracker

{

public static void main(String[] args)

{

float len, bre, area;

Scanner s = new Scanner(System.in);

System.out.print("Enter the Length of Rectangle: ");

len = s.nextFloat();

System.out.print("Enter the Breadth of Rectangle: ");

bre = s.nextFloat();

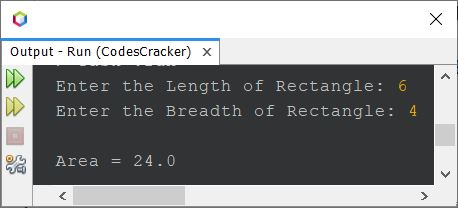
area = len\*bre;

System.out.println("\nArea = " +area);

}

}

The snapshot given below shows the sample run of above Java program with user input **6** as length and **4** as breadth of rectangle:



## Find Perimeter of Rectangle in Java

The question is, *write a Java program to find and print perimeter of a rectangle.* The answer to this question is the program given below:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

float length, breadth, perimeter;

Scanner s = new Scanner(System.in);

System.out.print("Enter the Length of Rectangle: ");

length = s.nextFloat();

System.out.print("Enter the Breadth of Rectangle: ");

breadth = s.nextFloat();

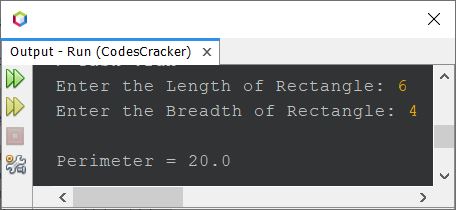
perimeter = (2\*length) + (2\*breadth);

System.out.println("\nPerimeter = " +perimeter);

}

}

The sample run of above program with same user input as of previous program's sample run, is shown in the snapshot given below:



## Find Area and Perimeter of Rectangle in Java - Single Program

This is the last program of this article, created basically to print area and perimeter both, of a rectangle, whose length and breadth will get entered by user at run-time of the program:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter the Length and Breadth of Rectangle: ");

float a = s.nextFloat();

float b = s.nextFloat();

float ar = a\*b;

float pr = 2\*(a+b);

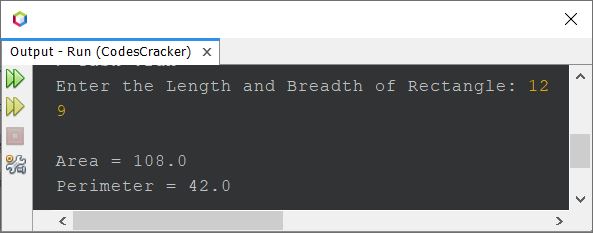
System.out.println("\nArea = " +ar);

System.out.println("Perimeter = " +pr);

}

}

Here is its sample run with user input **12** as length and **9** as breadth of rectangle:



## Celsius to Fahrenheit Conversion in Java

The question is, *write a Java program to convert Celsius to Fahrenheit. The Celsius value must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

class Example3

{

public static void main(String[] args)

{

float celsius, fahrenheit;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Temperature (in Celsius): ");

celsius = scan.nextFloat();

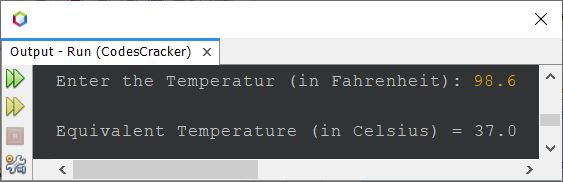
fahrenheit = (float) ((celsius\*1.8)+32);

System.out.println("\nEquivalent Temperature (in Fahrenheit) = " +fahrenheit);

}

}

The sample run of above program, with user input **98.6** as temperature in Fahrenheit, is shown in the snapshot given below:



## Celsius to Fahrenheit Conversion in Java

The question is, *write a Java program to convert Celsius to Fahrenheit. The Celsius value must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

class Example4

{

public static void main(String[] args)

{

float celsius, fahrenheit;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Temperature (in Celsius): ");

celsius = scan.nextFloat();

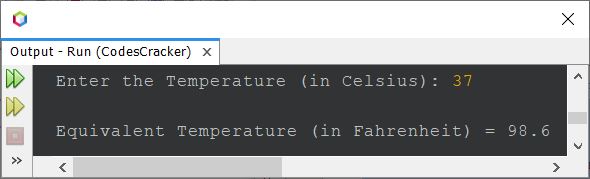
fahrenheit = (float) ((celsius\*1.8)+32);

System.out.println("\nEquivalent Temperature (in Fahrenheit) = " +fahrenheit);

}

}

The snapshot given below shows the sample run of above program with user input **37** as temperature in Celsius, to convert and print its equivalent Fahrenheit value:



# Java Program to Convert Days to Seconds

This article contains a program in Java to convert given number of days into seconds. There are 86400 seconds in a day.

import java.util.Scanner;

class Example5

{

public static void main(String[] args)

{

int noOfDays, noOfSeconds;

Scanner s = new Scanner(System.in);

System.out.print("Enter the Number of Days: ");

noOfDays = s.nextInt();

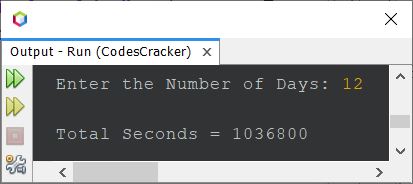
noOfSeconds = noOfDays\*86400;

System.out.println("\nTotal Seconds = " +noOfSeconds);

}

}

Here is its sample run with user input **12** as number of days to convert or find and print the total number of seconds passed in given 12 days:



This type of Java program comes under the category of simplest program, as here, we only need to receive the input from user and then multiply with **86400**. The result will be the total number of seconds in given number of days. So we only need to print the result value on output. That's it.

## Compute Simple Interest in Java

The question is, *write a Java program to compute simple interest based on principle, rate, and time period entered by user at run-time of the program.* The program given below is answer to this question:

import java.util.Scanner;

class Example6

{

public static void main(String[] args)

{

float p, r, t, si;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Principle Amount: ");

p = scan.nextFloat();

System.out.print("Enter the Rate of Interest: ");

r = scan.nextFloat();

System.out.print("Enter the Time Period (in Year): ");

t = scan.nextFloat();

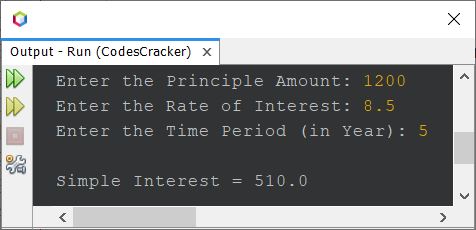
si = (p\*r\*t)/100;

System.out.println("\nSimple Interest = " +si);

}

}

Here is its sample run with user input **1200** as principle amount, **8.5** as rate of interest, and **5** as number of years or time period (in years):



# Java Program to Calculate Compound Interest

This article is created to cover a program in Java that find and prints compound interest based on the data provided by user at run-time of the program.

The formula to find compound interest is:

CI = (1 + r/n)(nt) - p

where **CI** indicates to Compound Interest, **p** indicates to Principal Amount, **r** indicates to the Annual Rate of Interest, **t** is Time Period for which the money is invested or borrowed, and **n** indicates to the Number of Times that the interest is compounded per unit **t**.

**Note -**If the interest is compounded monthly, then the value of **n** would be 12. If the interest is compounded quarterly, then the value of **n** would be 4, and so on. This happens when **t** is in years.

The question is, *write a Java program to find and print the compound interest based on the data such as****p****,****r****,****t****, and****n****values. The data must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

class Example7

{

public static void main(String[] args)

{

double p, r, t, n, ci;

Scanner s = new Scanner(System.in);

System.out.print("Enter the Value of p (Principal Amount): ");

p = s.nextDouble();

System.out.print("Enter the Value of r (Annual Rate of Interest): ");

r = s.nextDouble();

System.out.print("Enter the Value of t (Time Period): ");

t = s.nextDouble();

System.out.print("Enter the Value of n (Number of Times, Interest is Compounded): ");

n = s.nextDouble();

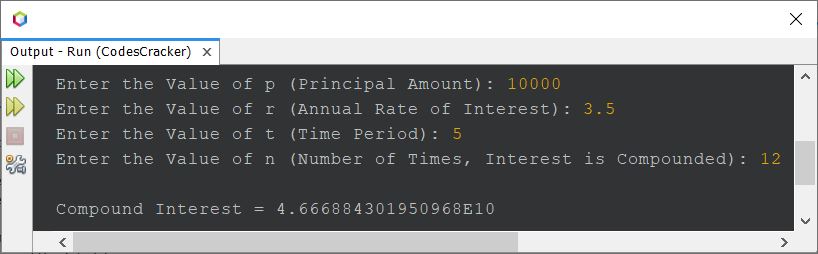
ci = p \* Math.pow(1 + (r/n), (n\*t)) - p;

System.out.println("\nCompound Interest = " +ci);

}

}

The snapshot given below shows the sample run of above Java program with user inputs **10000** as principal amount, **3.5** as rate of interest, **5** as time period, and **12** as number of times that the interest is compounded:



The above program can also be created in this way. This program prints the value of **Amount** too.

import java.util.Scanner;

public class Example8

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter the Value of p: ");

float p = s.nextFloat();

System.out.print("Enter the Value of r: ");

float r = s.nextFloat();

System.out.print("Enter the Value of t: ");

float t = s.nextFloat();

System.out.print("Enter the Value of n: ");

float n = s.nextFloat();

float amount = p \* (float)Math.pow(1 + (r/n), (n\*t));

float ci = amount - p;

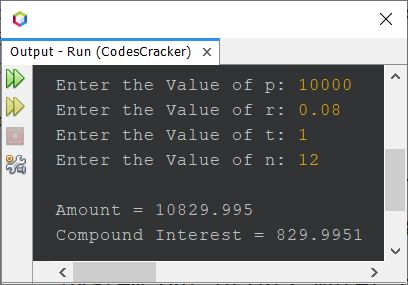
System.out.println("\nAmount = " +amount);

System.out.println("Compound Interest = " +ci);

}

}

Here is its sample run with user input **10000** as **p**, **0.08** as **r**, **1** as **t**, and **12** as **n**:



Examples on If Else Statements

## Largest of Two Numbers using if-else

The question is, *write a Java program to find largest between of two numbers. Both the number must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

class Codes

{

public static void main(String[] args)

{

int numberOne, numberTwo, largest;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the First Number: ");

numberOne = scan.nextInt();

System.out.print("Enter the Second Number: ");

numberTwo = scan.nextInt();

if(numberOne>numberTwo)

largest = numberOne;

else

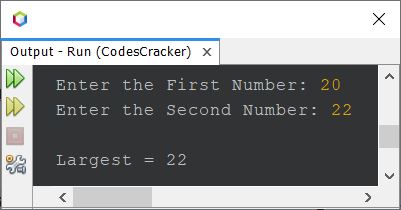
largest = numberTwo;

System.out.println("\nLargest = " +largest);

}

}

The snapshot given below shows the sample run of above Java program, with user input **20** and **22** as two numbers:



## Find Largest of Three Numbers using if else

The question is, *write a Java program to find the largest of three given numbers.* The program given below is its answer:

import java.util.Scanner;

class Codes

{

public static void main(String[] args)

{

int a, b, c, large;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the First Number: ");

a = scan.nextInt();

System.out.print("Enter the Second Number: ");

b = scan.nextInt();

System.out.print("Enter the Third Number: ");

c = scan.nextInt();

if(a>b && a>c)

large = a;

else if(b>a && b>c)

large = b;

else

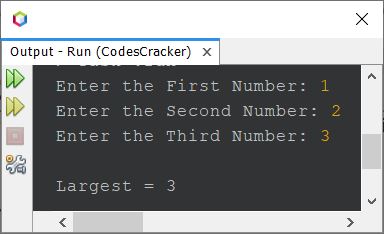
large = c;

System.out.println("\nLargest = " +large);

}

}

The snapshot given below shows the sample run of above program, with user input **1**, **2**, and **3** as first, second, and third number to find and print the largest among these three numbers:



## Check Odd or Even using if-else in Java

The question is, *write a Java program to check odd or even number. The number must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

public class Ex1

{

public static void main(String[] args)

{

int num;

Scanner scan = new Scanner(System.in);

System.out.print("Enter a Number: ");

num = scan.nextInt();

if(num%2==0)

{

System.out.println("\nIt is an Even Number.");

}

else

{

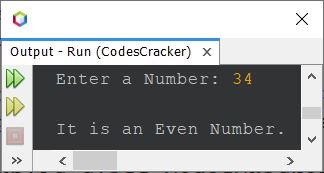
System.out.println("\nIt is an Odd Number.");

}

}

}

The snapshot given below shows the sample run of above Java program on checking whether a given number is an odd or an even number, with user input **34**



## Check Vowel or Consonant in Java - First Way

The question is, *write a Java program to check whether an input character (alphabet) is a vowel or consonant.* The program given below is its answer:

import java.util.Scanner;

class Ex2

{

public static void main(String[] args)

{

char ch;

Scanner scan = new Scanner(System.in);

System.out.print("Enter an Alphabet: ");

ch = scan.next().charAt(0);

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' ||

ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

System.out.println("\nIt is a Vowel.");

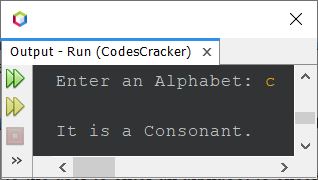
else

System.out.println("\nIt is a Consonant.");

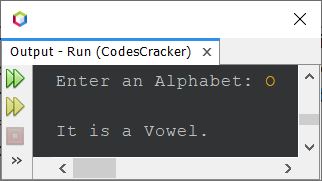
}

}

The snapshot given below shows the sample run of above Java program with user input **c** as an alphabet to check whether it is a vowel or not:



Here is another sample run with user input **O**, this time:



## Check Vowel or Consonant in Java - Second Way

This program does the same job as of previous program, but created using different way. This program uses character array to store all the 10 vowels. And further uses this array to compare and check whether the character entered by user is a vowel or not (consonant).

import java.util.Scanner;

class Ex3

{

public static void main(String[] args)

{

char ch;

int count=0;

char[] vowels = {'a','e','i','o','u','A','E','I','O','U'};

Scanner scan = new Scanner(System.in);

System.out.print("Enter an Alphabet: ");

ch = scan.next().charAt(0);

for(int i=0; i<10; i++)

{

if(ch==vowels[i])

{

count++;

break;

}

}

if(count==0)

System.out.println("\n" +ch+ " is a Consonant");

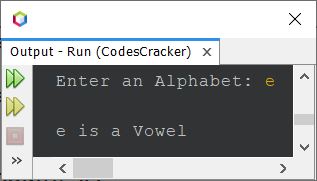
else

System.out.println("\n" +ch+ " is a Vowel");

}

}

Here is its sample run with user input **e**:



## Check Vowel or Consonant in Java - Complete Version

This is the last program of this article. I've called this program as the complete version of checking vowel or consonant, because this program handles with invalid character input too. That is, in previous programs, if user enters a character such as **2, $, }** etc., then the program prints the character is consonant. But this program does not. It only prints vowel or consonant, if the character is. Otherwise it prints the *character is neither vowel nor consonant*.

import java.util.Scanner;

class Ex4

{

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter an Alphabet: ");

char ch = scan.next().charAt(0);

int ascii = ch;

if(ch==65 || ch==69 || ch==73 || ch==79 || ch==85)

System.out.println("\n\'" +ch+ "\' is an Uppercase Vowel.");

else if(ch==97 || ch==101 || ch==105 || ch==111 || ch==117)

System.out.println("\n\'" +ch+ "\' is a Lowercase Vowel.");

else

{

if((ascii>=65 && ascii<=90) || (ch>=97 && ch<=122))

System.out.println("\n\'" +ch+ "\' is a Consonant.");

else

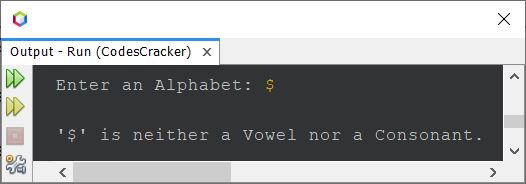
System.out.println("\n\'" +ch+ "\' is neither a Vowel nor a Consonant.");

}

}

}

Here is its sample run with user input **$**



**Note -**The ASCII values of **A-Z** are **65-90**. Whereas the ASCII values of **a-z** are **97-122**.

# Java Program to Check Alphabet

This post covers a program in Java that checks whether a character entered by user at run-time of the program, is an alphabet or not.

**Note -**All characters from either **A-Z** or **a-z** are alphabets.

## Check Alphabet using if-else in Java

The question is, *write a Java program to check whether a character is an alphabet or not using if-else.* The program given below is its answer:

import java.util.Scanner;

class Ex5

{

public static void main(String[] args)

{

char ch;

Scanner s = new Scanner(System.in);

System.out.print("Enter a Character: ");

ch = s.next().charAt(0);

if((ch>='A' && ch<='z') || (ch>='a' && ch<='z'))

{

System.out.println("\nIt is an Alphabet.");

}

else

{

System.out.println("\nIt is not an Alphabet.");

}

}

}

## Check Alphabet or Digit in Java

This program checks whether a given character is an alphabet, a digit, or any other character.

import java.util.Scanner;

class Ex6

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter a Character: ");

char ch = s.next().charAt(0);

if((ch>='A' && ch<='z') || (ch>='a' && ch<='z'))

System.out.println("\nThe character \'" +ch+ "\' is an Alphabet.");

else if(ch>='0' && ch<='9')

System.out.println("\nThe character \'" +ch+ "\' is a Digit.");

else

System.out.println("\nThe character \'" +ch+ "\' is neither an Alphabet nor a Digit.");

}

}

# Java Program to Calculate Discount and Price to Paid

This article is created to cover some programs in Java that can calculate the discount, and the price that has to be paid after providing the total amount of shopping. The discount must be applied based on the data given in following table:

|  |  |
| --- | --- |
| **Shopping Amount** | **Discount** |
| <=800 | No discount |
| >800 and <=1500 | 10% |
| >1500 and <=2500 | 15% |
| >2500 and <=5000 | 20% |
| >5000 | 30% |

That is, when the total shopping amount is less than 800, then there will no discount to apply. When the total shopping amount is greater than 800 and less than or equal to 1500, then there will be 10% discount to apply, and so on.

## Discount Program in Java

import java.util.Scanner;

class Ex7

{

public static void main(String[] args)

{

float totalCost, costToPaid, discount;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Total Amount of Shopping: ");

totalCost = scan.nextFloat();

if(totalCost<=800)

{

costToPaid = totalCost;

}

else if(totalCost>800 && totalCost<=1500)

{

discount = (totalCost\*10)/100;

costToPaid = totalCost - discount;

}

else if(totalCost>1500 && totalCost<=2500)

{

discount = (totalCost\*15)/100;

costToPaid = totalCost - discount;

}

else if(totalCost>2500 && totalCost<=5000)

{

discount = (totalCost\*20)/100;

costToPaid = totalCost - discount;

}

else

{

discount = (totalCost\*30)/100;

costToPaid = totalCost - discount;

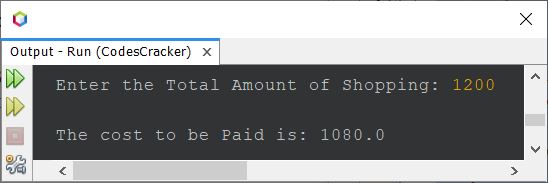
}

System.out.println("\nThe cost to be Paid is: " + costToPaid);

}

}

Here is its sample run with user input **1200** as total amount of shopping:



**Java Program to Check Leap Year or Not**

This article covers multiple programs in Java that checks whether an year entered by user at run-time of the program, is a leap year or not.

A leap year is an year that

* is divisible by 4, but not by 100
* or is divisible by 400

**Leap Year Program in Java**

The question is, *write a Java program that checks whether an year is a leap year or not. The year must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

class Ex8

{

public static void main(String[] args)

{

int year;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Year: ");

year = scan.nextInt();

if(year%4==0 && year%100!=0)

System.out.println("\nIt is a Leap Year.");

else if(year%400==0)

System.out.println("\nIt is a Leap Year.");

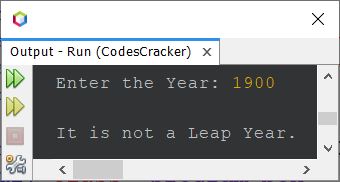
else

System.out.println("\nIt is not a Leap Year.");

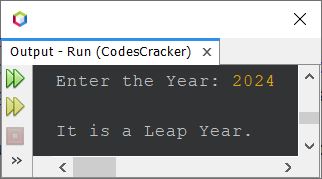
}

}

The snapshot given below shows the sample run of above program with user input **1900** as year to check whether it is a leap year or not:



Here is another sample run with user input **2024** as year:



**Check Leap Year using Conditional Operator in Java**

This program does the same job as of previous program. This program uses conditional or ternary (?:) operator to do the job.

import java.util.Scanner;

class Ex9

{

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Year: ");

int year = scan.nextInt();

int check = ((year%4==0 && year%100!=0) || (year%400==0)) ? 4 : 0;

if(check==4)

System.out.println("\n" +year+ " is a Leap Year.");

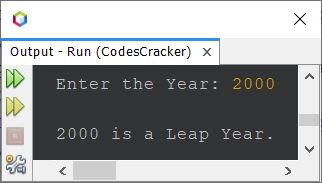
else

System.out.println("\n" +year+ " is not a Leap Year.");

}

}

Here is its sample run with user input **2000**:



The above program can also be created in this way:

import java.util.Scanner;

class Ex10

{

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Year: ");

int year = scan.nextInt();

String check = ((year%4==0 && year%100!=0) || (year%400==0)) ? "yes" : "no";

if(check.equals("yes"))

System.out.println("\n" +year+ " is a Leap Year.");

else

System.out.println("\n" +year+ " is not a Leap Year.");

}

}

## Telephone Bill Program in Java

The question is, *write a Java program to find and print the telephone bill that has to be paid for previous month, based on the total minutes of calls made.* Here is its answer. This program follows the rates as given above.

import java.util.Scanner;

public class Ex11

{

public static void main(String[] args)

{

int numberOfCalls;

float phoneBill;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Total Minutes of Calls Made this Month: ");

numberOfCalls = scan.nextInt();

if(numberOfCalls<=60)

phoneBill = 14;

else

{

numberOfCalls = numberOfCalls - 60;

phoneBill = 14 + (float)(numberOfCalls \* 0.12);

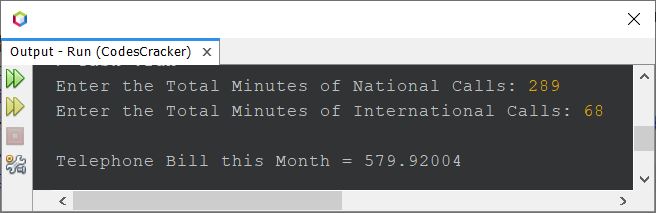
}

System.out.println("\nTelephone Bill this Month = " +phoneBill);

}

}

Here is its sample run with user input **432** as total minutes of calls made:



**Courier Charge based on Weight of the Parcel**

The question is, *write a Java program to compute courier charge based on parcel weight to ship. The charge based on weight must be calculated as per following criteria:*

* When the weight of parcel is less than or equal to 5 Kilogram, then the courier charge will be $6
* When the weight of the parcel is above 5 Kilogram, then there is an additional charge of $1.2 for each extra Kilogram

The program given below is the answer to this question:

import java.util.Scanner;

class Ex12

{

public static void main(String[] args)

{

float weight, courierCharge;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Weight of Parcel (in Kilogram): ");

weight = scan.nextFloat();

if(weight<=5)

{

courierCharge = 6;

}

else

{

weight = weight - 5;

courierCharge = (float)(6 + (weight\*1.2));

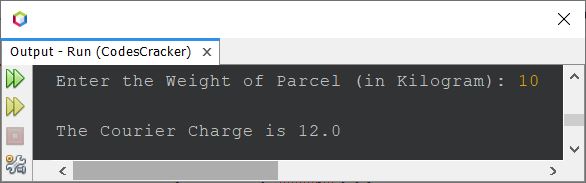
}

System.out.println("\nThe Courier Charge is " + courierCharge);

}

}

Here is its sample run with user input **10** as the weight of parcel (in Kg.) to ship:



## Calculator Program in Java using if else

The question is, *write a simple calculator program in Java that performs four basic arithmetic operations.* The program given below is its answer:

import java.util.Scanner;

class Ex13

{

public static void main(String[] args)

{

float a, b, res;

int choice;

Scanner scan = new Scanner(System.in);

System.out.println("1. Addition");

System.out.println("2. Subtraction");

System.out.println("3. Multiplication");

System.out.println("4. Division");

System.out.print("Enter Your Choice (1-4): ");

choice = scan.nextInt();

if(choice>=1 && choice<=4)

{

System.out.print("\nEnter any Two Number: ");

a = scan.nextFloat();

b = scan.nextFloat();

if(choice==1)

res = a+b;

else if(choice==2)

res = a-b;

else if(choice==3)

res = a\*b;

else

res = a/b;

System.out.println("\nResult = " +res);

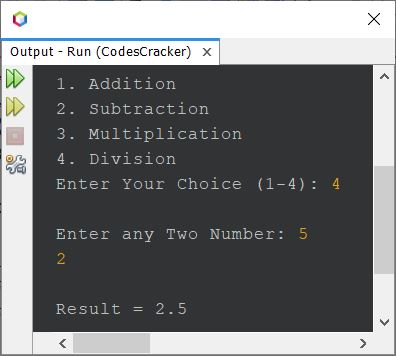
}

else

System.out.println("\nInvalid Choice!");

}

}



## Calculator Program in Java using switch

Here is another calculator program in Java created using **switch** case, instead of **if...else**. This program also continue its execution, until user enters **5** as choice to exit or stop the execution.

import java.util.Scanner;

class Ex14

{

public static void main(String[] args)

{

float a, b, res=0;

int choice;

Scanner scan = new Scanner(System.in);

while(true)

{

System.out.println("1. Addition");

System.out.println("2. Subtraction");

System.out.println("3. Multiplication");

System.out.println("4. Division");

System.out.println("5. Exit");

System.out.print("Enter Your Choice (1-5): ");

choice = scan.nextInt();

switch(choice)

{

case 1:

System.out.print("\nEnter any Two Number: ");

a = scan.nextFloat();

b = scan.nextFloat();

res = a+b;

break;

case 2:

System.out.print("\nEnter any Two Number: ");

a = scan.nextFloat();

b = scan.nextFloat();

res = a-b;

break;

case 3:

System.out.print("\nEnter any Two Number: ");

a = scan.nextFloat();

b = scan.nextFloat();

res = a\*b;

break;

case 4:

System.out.print("\nEnter any Two Number: ");

a = scan.nextFloat();

b = scan.nextFloat();

res = a/b;

break;

case 5:

return;

default:

System.out.println("\nInvalid choice!");

break;

}

System.out.println("\nResult = " +res+ "\n");

}

}

}

## Find Grades of Student in Java - Complete Version

* Since the program given above has some limitations such as what if user enters a mark, greater than 100 ?  
  also the program works on 8 subjects only ?  
  Therefore keeping in mind, these types of limitations, I've modified the program and created a new one as given below.

import java.util.Scanner;

class CodesCracker

{

public static void main(String[] args)

{

float[] marks = new float[8];

float sum=0;

Scanner scan = new Scanner(System.in);

System.out.print("Enter Total Number of Subjects: ");

int tot = scan.nextInt();

System.out.print("Enter Marks Obtained in " +tot+ " Subjects: ");

for(int i=0; i<tot; i++)

{

marks[i] = scan.nextFloat();

sum += marks[i];

}

float avg = sum/tot;

String grade = "";

if(avg>=94 && avg<=100)

grade = "A";

else if(avg>=90 && avg<94)

grade = "A-";

else if(avg>=87 && avg<90)

grade = "B+";

else if(avg>=83 && avg<87)

grade = "B";

else if(avg>=80 && avg<83)

grade = "B-";

else if(avg>=77 && avg<80)

grade = "C+";

else if(avg>=73 && avg<77)

grade = "C";

else if(avg>=70 && avg<73)

grade = "C-";

else if(avg>=67 && avg<70)

grade = "D+";

else if(avg>=63 && avg<67)

grade = "D";

else if(avg>=60 && avg<63)

grade = "D-";

else if(avg>=0 && avg<60)

grade = "F";

else

grade = "Invalid Input!";

if(grade.equals("Invalid Input!"))

System.out.println("\nInvalid Input!");

else

System.out.println("\nGrade = " +grade);

}

}

Here is its sample run with user input **5** as number of subjects and **70, 78, 90, 94, 88** as five subject marks: