

***** OUTPUT *****

So far this is only a representation of getting the roman numeral entered by the end user and presenting this on the screen.

There is validation to ensure that correct numerals are presented.

Also, it will complete a conversion of a number consisting of single roman numeral.

Challenges appear in coding once it extends beyond single roman numeral.

The simple part is two identical numerals in a whole roman numeral such as XX, CC.

```
Welcome to Online IDE!! Happy Coding :)
Enter number numerals in the number to be converted to decimal:

2
Enter roman numeral 1 of 2:

V
Enter roman numeral 2 of 2:

V
This will be converted to decimal:VV
Conversion is:10

** Process exited - Return Code: 0 **
```

Example showing correct numeral but syntax incorrect.
This is to be explored further into code logic.

```
Welcome to Online IDE!! Happy Coding :)
Enter number numerals in the number to be converted to decimal:

3
Enter roman numeral 1 of 3:
F
Please input a successful roman numeral: M D C L X V I
Enter roman numeral 1 of 3:
G
Please input a successful roman numeral: M D C L X V I
Enter roman numeral 1 of 3:
C
Enter roman numeral 2 of 3:
M
Enter roman numeral 3 of 3:
C
This will be converted to decimal:CMC
```

*** CODE ***

```
import java.util.Scanner;
import java.util.Arrays;
/*
Online Java - IDE, Code Editor, Compiler

Online Java is a quick and easy tool that helps you to build, compile, test your programs online.
*/
public class Main
{
    public static void main(String[] args) {

        System.out.println("Welcome to Online IDE!! Happy Coding :)");

        char[] acceptedNumerals = new char[]{'M','D','C','L','X','V','I'};
        //These are all accepted roman numerals

        int number; //This is set once end user specifies number numerals in their number to be inputted

        boolean noMatch=true; // this flag is used to check if the numeral entered is a valid one

        int j; //used to increment the character array containing user input of numeral
        int count=0; // this keeps track number times end user has entered a character. It can be decremented if it
        fails validation.

        //int pos;

        Scanner reader=null;

        // This is used to ascertain how many digits end user wants
        reader = new Scanner(System.in); // Reading from System.in
        System.out.println("Enter number numerals in the number to be converted to decimal:");
        number=reader.nextInt();

        char inputtedNumerals[] = new char[number]; // this creates character array with end user specified size
        int [][] posValues = new int [number][2]; // this is used to keep track of:

        // this is 2D array. 'number' simply increments with array index of the roman numeral in character array
        // the multidimensional aspect at 0 index will store: index of which numeral i.e M is 0, D is 1
        //aspect at 1 index will store: decimal conversion
        // This will support later analysis.

        do
        {
            reader = new Scanner(System.in); // Reading from System.in
            System.out.println("Enter roman numeral " + (count+1) + " of " + number + ":"); // end user prompted to enter
            roman numeral
            inputtedNumerals[count]=reader.next().charAt(0);
            count++;

            for (j=0; j<inputtedNumerals.length;j++) // this will check each item in the inputted area against 7 roman
            numerals.

                // the condition could equally be set to count for better loop efficiency if the
                //number entered is long

            {

                for (int k=0; k<acceptedNumerals.length;k++) // this goes through roman numerals
                {

                    if (inputtedNumerals[j]==acceptedNumerals[k]) // this checks to see if the roman numerals entered are valid
```

```

{

    posValues[count-1][0]=k;

    // This switch statement now checks 'k' valid position in hierarchy of the 7 numerals.
    // 'M','D','C','L','X','V','I'

    switch(k)
    {
        case 0:
            posValues[count-1][1]=1000;
            break;

        case 1:
            posValues[count-1][1]=500;
            break;

        case 2:
            posValues[count-1][1]=100;
            break;

        case 3:
            posValues[count-1][1]=50;
            break;

        case 4:
            posValues[count-1][1]=10;
            break;

        case 5:
            posValues[count-1][1]=5;
            break;

        case 6:
            posValues[count-1][1]=1;
            break;
    }

    //pos=k;

    noMatch=false; // This sets flag stating match has been found.
    k=acceptedNumerals.length-1;
    break;

}

}
// This will now give information if the user input is valid for each numeral.
// If found to be incorrect, the count will go back and end user will need to enter it again.

if (noMatch==true)
{
    System.out.println("Please input a successful roman numeral: M D C L X V I");
    count=count-1;
    break;
}

}

} while (count<inputtedNumerals.length);

String str = String.valueOf(inputtedNumerals);

```

```
System.out.println("This will be converted to decimal:" + str); // This informs end user of the roman numerals to be converted
```

```
performConversion pc = new performConversion(acceptedNumerals, inputtedNumerals,posValues); // calls constructor in class to perform conversion  
}  
}
```

```
class performConversion
```

```
{  
    char[] inputtedNumerals;  
    char[] acceptedNumerals;  
    int [][] posValues;  
  
    // This is the constructor used to call the converter method.  
    public performConversion (char[] aNumerals, char [] inputNumerals, int [][] pValues)  
    {  
        this.acceptedNumerals=aNumerals;  
        this.inputtedNumerals=inputNumerals;  
        this.posValues=pValues;  
        converter();  
    }  
}
```

```
public void converter()
```

```
{  
    // There will be NO issues in performing conversion of a single roman numeral
```

```
    if (inputtedNumerals.length==1) // This functions  
    {  
        System.out.println("Conversion is:" + posValues[0][1]);  
    }  
}
```

```
    // this is 2D array. count simply increments with array index of the roman numeral in character array  
    // the multidimensal aspect at 0 index will store: index of which numeral i.e M is 0, D is 1  
    //aspect at 1 index will store: decimanl conversion
```

```
    // ONCE ALL PRINCIPLES ARE UNDERSTOOD FOR A VALID ROMAN NUMERAL, THE CODE WILL BE IMPLEMENTED FURTHER
```

```
        if (inputtedNumerals.length==2)  
        {  
            if (posValues[0][0]==posValues[1][0])  
            {  
  
                System.out.println("Conversion is:" + (posValues[0][1] + posValues[1][1]));  
            }  
        }  
    }  
}
```