

## Cobol's Hello World

```
identification division.  
program-id. HelloWorld.  
procedure division.  
display "Hello World!" .
```

## Java's Hello World:

```
Class HelloWorld {  
public static void main(String[] args) {  
System.out.println( "Hello World!" );  
}  
}
```

## Cobol's Sum of Integers up to Input

```
identification division.  
program-id. sumofintegers.  
data division.  
working-storage section.  
01 n binary-int.  
01 i binary-int.  
01 sum binary-int.  
procedure division.  
display "Enter a positive integer"  
accept n  
perform varying i from 1 by 1 until i>n  
add i to sum  
end-perform  
display "The sum is " sum.
```

## Java's Sum of Integers up to Input

```
import java.util.Scanner;  
public class sumofintegers {  
public static void main(String[] arg) {  
System.out.println( "Enter a positive integer" );  
Scanner scan=new Scanner(System.in);  
int n=scan.nextInt();  
int sum=0;  
for (int i=1;i<=n;i++) {  
sum=sum+i;  
}  
System.out.println( "The sum is "+sum);  
}
```

## Emacs' Sum of Integers, `interactive' Version

```
(defun get-integer-for-summing (integer)  
"Add up all numbers from 1 up to integer"  
(interactive "nEnter a positive integer: ")  
(loop for number from 1 to integer  
sum number into answer  
finally return  
(message "The sum to %s is %s" integer answer)))
```

get-integer-for-summing

Brian Small made this page with emacs org-mode on a gnu/Linux operating system.

## Emacs-Lisp's Hello World

```
(message "Hello World")
```

Hello World

## Emacs Lisp's Hello World 2:

```
(princ "Hello World")
```

Hello World

## Emacs Sum of Integers up to Input, `while' Version

```
(setf integer  
(string-to-number  
(read-string "Enter a positive integer: " nil nil "10")  
(let ((i integer) (sum 0))  
(while (> i 0)  
(setf sum (+ i sum))  
(setf i (- i 1)))  
(message "The sum to %s is %s" integer sum))
```

The sum to 15 is 120

## Emacs' Sum of Integers up to Input, `loop' Version

```
(setf integer  
(string-to-number  
(read-string "Enter a positive integer: " nil nil "10")  
(loop for number from 1 to integer  
sum number into answer  
finally return  
(message "The sum to %s is %s" integer answer))
```

The sum to 15 is 120

## Emacs is Great

Let's use Emacs for education.

1. Emacsテクニックバイブル ～作業効率をカイゼンする200の技～
  - 技術評論社 (2010/8/3)
2. Emacs Lispテクニックバイブル
  - 技術評論社 (2011/11/26)
3. Emacs実践入門 ～思考を直感的にコード化し、開発を加速する (WEB+DB PRESS plus)
  - 技術評論社 (2012/3/7)

Cobol and Java Source from GNU COBOL 2.0

Programmers Guide <http://opencobol.add1tocobol.com/gnucobol/>

<http://opencobol.add1tocobol.com>

/GNU%20Cobol%202.0%20Programmers%20Guide.pdf

```

#+TITLE: Cobol vs. Emacs Lisp
#+OPTIONS: toc:nil num:nil

#+HTML_HEAD: <style> @page { size: 210mm 297mm; margin: 4mm; margin-bottom: 6mm; } </style>
#+HTML_HEAD: <style> h1.title { display:none; } </style>
#+HTML_HEAD: <style> h2 { font-size: 9pt; } </style>
#+HTML_HEAD: <style> div.outline-2 { width: 48%; float: left; } </style>
# #+HTML_HEAD: <style> pre.src { box-shadow: none; } </style>
#+HTML_HEAD: <style> pre.src { border: none; } </style>
#+HTML_HEAD: <style> div#postamble { display: none; } </style>

```

```

** Cobol's Hello World

```

```

#+name: HelloWorldCobol
#+BEGIN_SRC Cobol
identification division.
program-id. HelloWorld.
procedure division.
display "Hello World!" .
#+END_SRC
#+name: HelloWorldJava
#+BEGIN_SRC Java
Java:
Class HelloWorld {
public static void main(String[] args) {
System.out.println( "Hello World!" );
}
}
#+END_SRC

```

```

** Emacs-Lisp's Hello World

```

```

#+name: HelloWorldelmessage
#+BEGIN_SRC emacs-lisp
(message "Hello World")
#+END_SRC
#+RESULTS:
: Hello World

```

```

#+name: HelloWorldprinc
#+BEGIN_SRC emacs-lisp :exports both
(princ "Hello World")
#+END_SRC
#+RESULTS: HelloWorldprinc
: Hello World

```

```

#+HTML: <br style="clear:both;" />

```

```

** Cobol's Sum of Integers up to Input

```

```

#+name: SumOfIntegersFromInputCob
#+BEGIN_SRC Cobol
identification division.
program-id. sumofintegers.
data division.
working-storage section.
01 n binary-int.
01 i binary-int.
01 sum binary-int.
procedure division.
display "Enter a positive integer"
accept n
perform varying i from 1 by 1 until i>n
add i to sum
end-perform
display "The sum is " sum.
#+END_SRC
** Emacs Sum of Integers up to Input, `while' Version
Emacs Lisp
#+name: SumOfIntegersFromInputWhileLoop
#+BEGIN_SRC emacs-lisp :exports both :eval no
(setf integer
  (string-to-number
    (read-string "Enter a positive integer: " nil nil "10")))
(let ((i integer) (sum 0))

```

```

    (while (> i 0)
      (setf sum (+ i sum))
      (setf i (- i 1)))
    (message "The sum to %s is %s" integer sum))
#+END_SRC
With Results
#+RESULTS: SumOfIntegersFromInputWhileLoop
: The sum to 15 is 120

```

```

** Java's Sum of Integers up to Input
#+name: SumOfIntegersFromInputJava
#+BEGIN_SRC Java
import java.util.Scanner;
public class sumofintegers {
public static void main(String[] arg) {
System.out.println( "Enter a positive integer" );
Scanner scan=new Scanner(System.in);
int n=scan.nextInt();
int sum=0;
for (int i=1;i<=n;i++) {
sum=sum+i;
}
System.out.println( "The sum is "+sum);
}
}
#+END_SRC

```

```

** Emacs' Sum of Integers up to Input, `loop' Version
Emacs Lisp
#+name: SumOfIntegersFromInputElLoop
#+BEGIN_SRC emacs-lisp :exports both :eval no
  (setf integer
    (string-to-number
      (read-string "Enter a positive integer: " nil nil "10")))
  (loop for number from 1 to integer
    sum number into answer
    finally return
      (message "The sum to %s is %s" integer answer))
#+END_SRC
#+RESULTS: SumOfIntegersFromInputElLoop
: The sum to 15 is 120

```

```

** Emacs' Sum of Integers, `interactive' Version
#+name: SumOfIntegersFromInputElLoop2
#+BEGIN_SRC emacs-lisp :exports both :eval no
  (defun get-integer-for-summing (integer)
    "Add up all numbers from 1 up to integer"
    (interactive "nEnter a positive integer: ")
    (loop for number from 1 to integer
      sum number into answer
      finally return
        (message "The sum to %s is %s" integer answer)))
#+END_SRC

```

```

#+RESULTS: SumOfIntegersFromInputElLoop2
: get-integer-for-summing

```

The Emacs editor and Lisp(List Processing) language seems very good for education. The 'Scheme' and 'Racket'

\*\* Emacs is Great

I recommend Emacs for education.

1. Emacsテクニックバイブル ～作業効率をカイゼンする200の技～  
- 技術評論社 (2010/8/3)
2. Emacs Lispテクニックバイブル  
- 技術評論社 (2011/11/26)
3. Emacs実践入門 ～思考を直感的にコード化し、開発を加速する (WEB+DB PRESS plus)  
- 技術評論社 (2012/3/7)

Cobol and Java Source from GNU COBOL 2.0 Programmers Guide

<http://opencobol.add1tocobol.com/gnucobol/>

<http://opencobol.add1tocobol.com/GNU%20Cobol%202.0%20Programmers%20Guide.pdf>