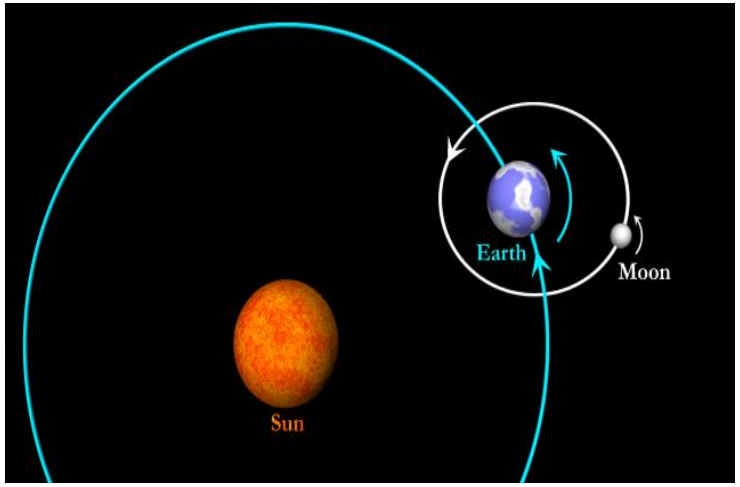
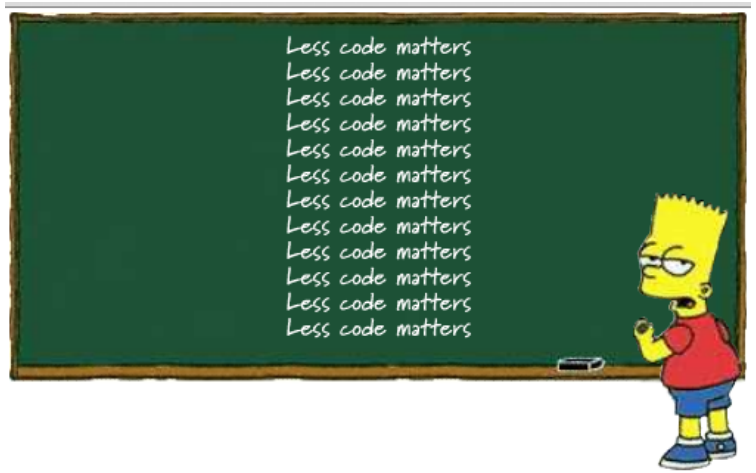
Loops

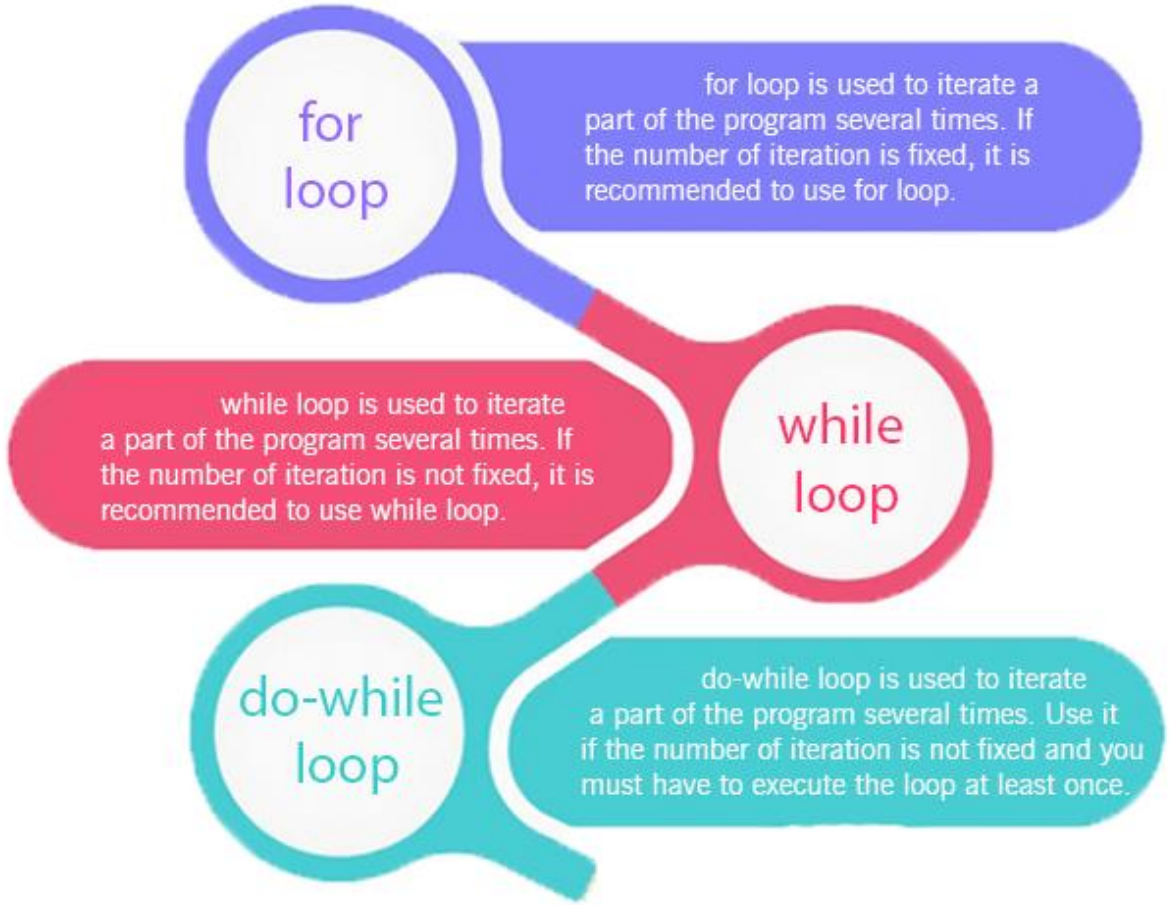
HK Rana


Earth
Moon
Sun


Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters
Less code matters

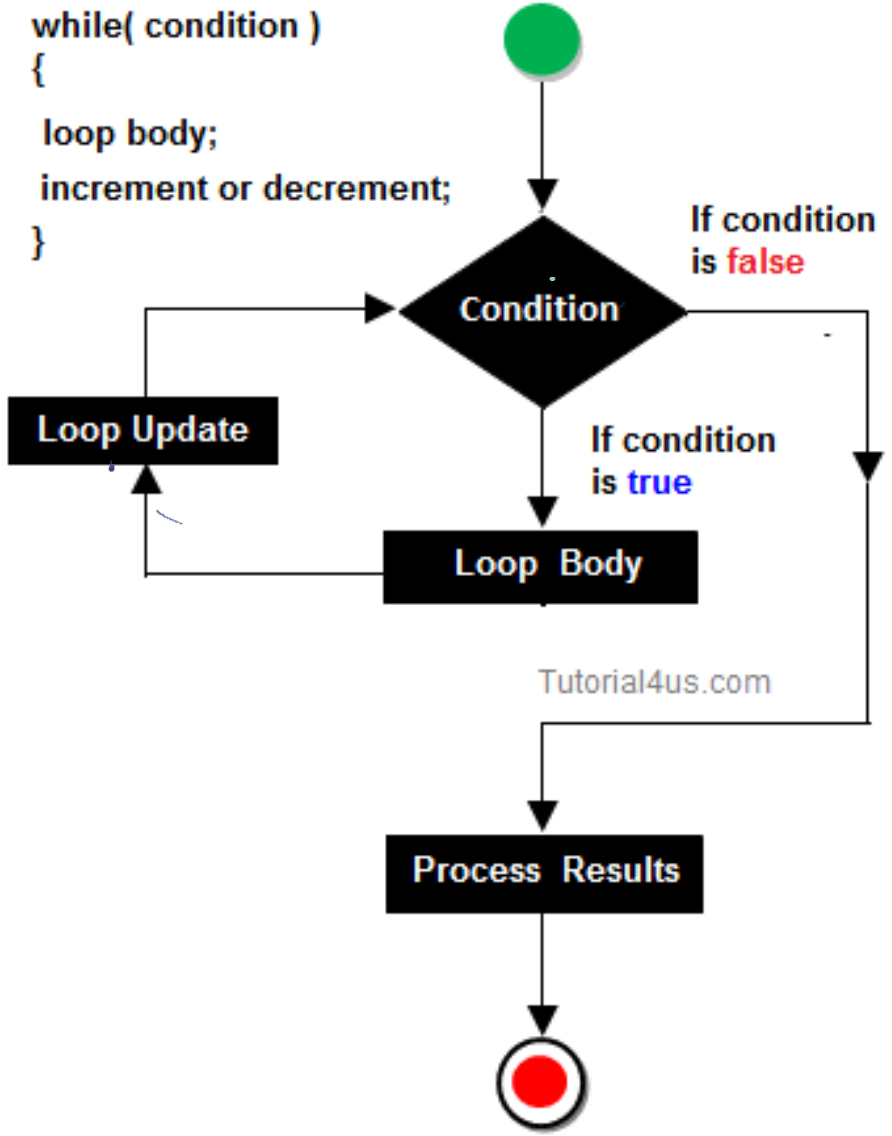
Why Looping??

Various Loops



Flow Chart of Loops:

for vs while vs do-while



PROGRAM Printing a Table of Squares

Let's write a program that prints a table of squares. The program will first prompt the user to enter a number *n*. It will then print *n* lines of output, with each line containing a number between 1 and *n* together with its square:

```
This program prints a table of squares.
Enter number of entries in table: 5
          1         1
          2         4
          3         9
          4        16
          5        25
```

PROGRAM **Summing a Series of Numbers**

As a second example of the `while` statement, let's write a program that sums a series of integers entered by the user. Here's what the user will see:

Infinite Loop

PROGRAM Calculating the Number of Digits in an Integer

Although the `while` statement appears in C programs much more often than the `do` statement, the latter is handy for loops that must execute at least once. To illustrate this point, let's write a program that calculates the number of digits in an integer entered by the user:

```
Enter a nonnegative integer: 60
The number has 2 digit(s).
```

for statement Idioms & Comma Separators

- *Counting up from* 0 *to* n–1*:*

idiom
```
for (i = 0; i < n; i++) …
```

- *Counting up from* 1 *to* n*:*

idiom
```
for (i = 1; i <= n; i++) …
```

- *Counting down from* n–1 *to* 0*:*

idiom
```
for (i = n - 1; i >= 0; i--) …
```

- *Counting down from* n *to* 1*:*

idiom
```
for (i = n; i > 0; i--) …
```

Nested Loop

Exiting from a loop

THE BREAK STATEMENT

THE CONTINUE STATEMENT

THE GOTO STATEMENT

PROGRAM

Balancing a Checkbook

Many simple interactive programs are menu-based: they present the user with a list of commands to choose from. Once the user has selected a command, the program performs the desired action, then prompts the user for another command. This process continues until the user selects an "exit" or "quit" command.