

CMS 10

11-17-08

11

RECALL: = IS THE ASSIGNMENT OPERATOR IN C++

== IS THE COMPARE-FOR-EQUALITY OPERATION

CLASSIC ERROR: USE = WHEN YOU MEAN ==

THE ASSIGNMENT

variable = expression;

IS ITSELF AN EXPRESSION, WITH A VALUE. THE VALUE OF THIS EXPRESSION IS THE VALUE ASSIGNED.

EX. $a = (b = c);$

is equivalent to

$b = c;$

$a = b;$



FURTHERMORE! IF A NUMERIC VALUE IS SUPPLIED WHERE `bool` IS EXPECTED, THEN THE NUMERIC VALUE IS CONVERTED TO `bool`!

NON-ZERO NUMERIC VAL. \rightarrow true

ZERO NUMERIC VAL. \rightarrow false

ITERATIVE DEFINITIONS

- while
- do-while
- for

while (cond) or while (cond) {

 stmt;

 .

 .

 .

 stmt;

 .

 .

 stmt;

}

 stmt;

 .

 .

RECALL:

PSEUDO-CODE

- 1.) get n
- 2.) $sum \leftarrow 0$
- 3.) $i \leftarrow 1$
- 4.) while $i \leq n$
- 5.) $sum \leftarrow sum + i$
- 6.) $i \leftarrow i + 1$
- 7.) Print sum

74

C++

```
int n, i, sum;
```

```
cin >> n;
```

```
sum = 0;
```

Initialize
L.C.V.

```
i = 1;
```

```
while (i <= n) {
```

TEST I.R.C.

```
sum = sum + i;
```

```
i = i + 1;
```

INCREMENT
L.C.V.

```
}
```

```
cout << sum << endl;
```

```

do
  stmt;
while (cond);
stmt;
...
} while (cond);
stmt;
...

```

```

Ex.
int n, i, sum;
cin >> n;
sum = 0;
i = 1;
do {
  sum = sum + i;
  i = i + 1;
} while (i <= n);
cout << sum << endl;

```

```

for (initialize LCV; TEST LRC; INCREMENT LCV) {
    start;
    .
    .
    start;
}
start;

```

EX.

```

int n, i, sum;
cin >> n;
sum = 0;
for (i = 1; i <= n; i = i + 1)
    sum = sum + i;
cout << sum << endl;

```

Compound Assignment ops.

- $a = a + b;$ $a += b;$
- $a = a - b;$ $a -= b;$
- $a = a * b;$ $a *= b;$
- $a = a / b;$ $a /= b;$

Ex. for ($i=1; i<=n; i++$)
 Sum += i;

AUTO INCREMENT & AUTO DECREMENT AS

Postfix Prefix

$a = a + 1;$ $a + 1;$ $++a;$

$a = a - 1;$ $a - 1;$ $--a;$

$c = (a + 1);$ or $c = (++a);$

Ex. $int\ a = 1, c;$

