Class 2: Conditionals and Loops

Sunday, April 6, 2025 10:11 AM

- → Logical operators
- → Precedence of operators
- → More on conditionals
- → Loops → for
  - → while
  - → do-while

Logical Operators

↳ [&&, ||, !]

&& (AND)

T	T	T
T	F	F
F	T	F
F	F	F

(5 > 3) && (10 <= 11)

T && T

T

(11 < 9) && (13 > 11)

F && T

F

short circuiting

|| (OR)

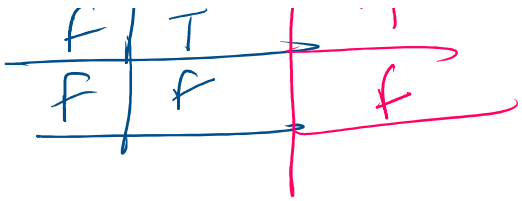
T	T	T
T	F	T
F	T	T
F	F	F

e) (21 < 13) || (9 >= 11)

f || f = f

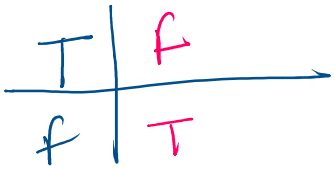
e) (11 >= 11) || (9 < 11)

T = T



e) 11>=11 || 12<11
T
T = T
Short circuiting

! (NOT)



e) !(11<=13)
!T = f

e) 5 x 3/2 + 5-3
+ 5x1 + 5-3
BODMAS 5+5-3
10-3
= 7

= [A) 9]
B) 7
C) 5

15/2 + 5-3
= 7 + 5-3 = 9

5 x 3/2

Precedence of operators
→ (%), (DM), (AS)] Mathematical

e) (5%3x2) + (11-7x3)

$(2 \times 2) + (11 - 21)$

$4 + \quad -10 \Rightarrow -6$

→ $!0 > 88 > 11$ logical operators

a) $(\,(5 >= 3)\ \&\&\ (13 > 11)\ ||\ (3 > 1)\,)$

T && F || T

F || T ⇒ T

more on conditionals

→ else is always associated with nearest if

Ex:

```
if(  ) {
}
```

```
if(  )
else
```

Loops → Repeat a Programming statement

→ for

- for
- while
- do while

for loop

for(initialization; condition; updation)

```
for(int i=1 ; i<=10 ; i=(i+1)){
    cout<<"Hello";
}
```

Hello
Hello
.
.
10

i=1 Hello
i=2 Hello
i=3 Hello
.
.
i=10 Hello
i=11

```
for(int i=1 ; i<=5 ; i=(i+2)){
    cout<<(i);
}
```

i=1 → 1
i=3 → 3
i=5 → 5
i=7 ✗

(i+2)
i=3+2
i=7

i=2

Q) Print table of 2 using loops

2 4 6 8 10 ... 20

2 4 6 8 10 ... 20

```
for( int i=2; i<=20; i=(i+2) {
    sout (i);
}
```

```
for (i=1; i<=10; (=(i+1) ) {
    sout (i*2);
}
```

i=2
(2)
i=2 2
i=3 4
i=4 8

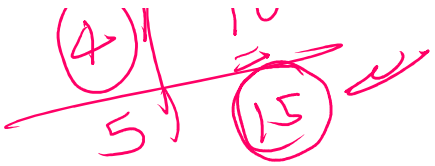
Q) Sum of first 5 natural numbers

Sol:

```
int sum = 0;
for(int i=1; i<=5; i++) {
    sum = sum + i;
}
```

```
int sum=0;
for (int i=1; i<=5; i++) {
    sum = sum + i;
}
sout (sum)
```

i	sum
1	1
2	3
3	6
4	10




```
int sum = 0
for (int i = 1; i <= 3; i++) {

}
```

Q) Factorial of a number.

$$N! = 1 \times 2 \times 3 \cdots \times N$$

Eg: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$

Sol:

```
int ans = 1;
for (int i = 1; i <= 5; i++) {
    ans = ans * i;
}
```

ans: ~~1~~ ~~2~~ ~~6~~ ~~24~~ 120

i = 1
i = 2
i = 3
i = 4
i = 5
i = 6

Reverse loops

Q) Print 10-1 using loops (10, 9, 8 ---- 1)

Sol:

```
for (int i = 10; i >= 1; i = i - 1) {
    i--
```

...able of 3 in reverse order.

o) Print table of 3 in reverse order.

30, 27, 24 --- 3

-3

```
for( int i= 30; i>= 3; i= i-3)
    sout (i);
}
```

for (init ; condition ; updation) {

}

while loop

```
init
while (condition) {
    updation
}
```

o) `for(int i=1; i<=10; i++) { sout ("Hello"); }`

```
int i=1;
while (i<=10) {
    sout ("Hello");
    i++;
}
```

```
int i=1;
while (i<=5) {
    sout ("Hello");
}
```

i=1	Hello
i=2	Hello
i=3	Hello
i=4	Hello

```
while (i<=10)
  sout ("Hello");
  i++;
}
```

```
i=4
i=5
i=6
```

Hello
Hello

Q) Print Table of 7 in both normal and reverse order using while loop.

Soln Normal : 7, 14, 21 ... 70

```
int i=7;
while (i<=70)
  sout (i);
  i=i*7;
```

Reverse 70, 63, 56 ... 7.

```
int i=70;
while (i>=7)
  sout(i);
  i=i-7;
}
```

→

```
i=10;
while (i>=1)
  sout (i*7);
  i=i-1;
}
```

do - while

```
int
do {
  logic
  updation
} while (condition);
```

→

```
int i=7;
do {
  sout (i)
  i=i+7;
} while (i<=70);
```

→ int
do {
logic
updation
} while (condition);

→

```
int i = 7;
do {
    sout(i)
    i = i+1;
} while (i<=70);
```

④ do-while definitely executes min of 1 iteration.

Show off knowledge

→ Sbyte → 1 Byte
→ short → 2 Byte
→ int → 4 Byte → 32 bits
→ long → 8 Byte → 64 bits
→ float → 4 Byte → 32 bits
→ double → 8 Bytes → 64 bits
→ char → 2 Bytes → 16 bits.

bit (1)(0)(1)

8 bit = 1 bytes
1 kB = 1024 bytes
1 mB = 1024 kB
1 GB = 1024 mB

{2 bit
$2^2 = 4$ {
0 0
0 1
1 0
1 1

1/0
(⊖)(⊖)
$2 \times 2 = 4$

4 bits
2^4
[(⊖) - - -
$2 \times 2 \times 2 \times 2$]

n bits → 2^n different numbers

$[-2^{31} - (2^{31}-1)]$ 32 bits → 2^{32}

$[0 - 2^{32}]$

$2^{32} \rightarrow$ +ve, −ve, 0

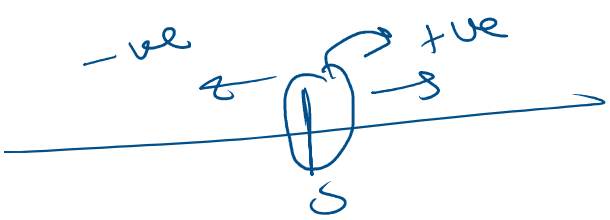
2^{31} 2^{31}

$\Rightarrow 2^{31}$

$-2^{31} - 2^{31}$

$-5 - (4)$

(5) ≥ 5



→ 8 bytes = 64 bits ⇒ $\frac{2^{64}}{2} = ($

$[-2^{63} - (2^{63}-1)]$

→ int : $[-10^9 \Rightarrow 10^9]$

→ ly $[-10^{18} - 10^{18}]$