Lecture - 3

Operators:

1. Binary operator: + , - , / , * , %

Example:

```
int main()
{
int a = 5, b= 3;
printf("%d\t%d",a+b,a-b);
return 0;
}
```

```
8	2
```

2. Unary operator: + ,-

Example:

```
int main() {
int a = 5, b= 3;
printf("%d",-a+b);
return 0;
}
```

```
-2
```

3. Relational operator: > , > = ,< , < = , = = , ! =
   true will return 1 and false will return 0.

Example:

```
int main()
{
int a = 5, b= 3;
printf("%d%d",a>b,a= =b);
return 0;
}
```

```
10
```

4. Logical operator: && (AND) , || (OR) , ! (NOT)
   true will return a nonzero value and false will return 0.

Example:

```
void main()
{
int a = 5, b= 3;
printf("%d%d",(a>b)&&(a!=b),!(b>a)| |(b= =a));
}
```

```
11
```

5. Bitwise operator: & (AND), | (OR), ^ (XOR) , << , >> , ~

A	b	A AND b
0	0	0
0	1	0
1	0	0
1	1	1

x

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

+

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

same hoile 0

a	NOT a
0	1
1	0

!

Example:

```
void main()
{
int a = 40, b= 15;
printf("%d", ~(((a>>2) & (b<<3)) ^ (a | b)));
}
```

-40

6. Increment and decrement operator: + + , - -

Example:

```
void main()
{
int i = 5;
printf("%d%d",i++,i--);
printf("%d",i);
}
```

455

7. Assignment operator: = , + = , * = , - = , / = , % = , & = , | = , ^ = , << = , >> =

Example:

```
void main()
{
int a=5,b=3;
a+=b; b*=a;
printf("%d%d",a,b);
}
```

824

Punctuators: ` @ # $ ( ) { } [ ] ; : " ' , . ?