

(local conv)

Type Casting

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main ( )
```

```
{ int a, b;
```

```
float c;
```

```
clrscr();
```

```
printf("Enter 2 int:");
```

```
scanf("%d %d", &a, &b);
```

```
c = (float) a / b;
```

```
printf("Div is %f", c);
```

```
getch();
```

$c = a(\text{float}) / b; \text{X}$

~~$c = a / b;$~~

$c = (\text{float})(a / b);$

$c = (\text{float}) a / b;$

OR

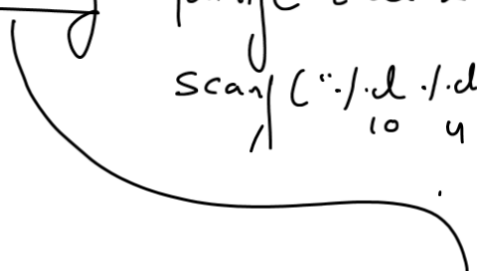
$c = a / (\text{float}) b;$

Syntax:

- ① (data type) var ;
- ② (data type) const ;
- ③ (data type) (expr) ;

(local conv) Type Casting

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b;
    printf("Enter 2 int:");
    scanf("%d %d", &a, &b);
    printf("Div is %.2f", (float)a/b);
    getch();
    clrscr();
}
```



int a = 5;

X printf("%f", a);
Observed answer

✓ printf("%f", (float)a);
5.000000

① avg = sum / 3.0;

or

avg = sum / (float) 3;

Type casting
of const!

② avg = (a+b+c) / 3.0;
or
avg = (float)(a+b+c) / 3;

Type casting of
exp

```
void main()
{
    float a, b;
    int c;
```

```
    clrscr();
    printf("Enter 2 floats:");
    scanf("%f %f", &a, &b);
    // 10.9 3.6
```

```
    c = a / b; X
    c = (int) a / (int) b;
    printf("Rem is %d", c);
    getch();
```

```
int x = 65;
```

```
printf("%d", x);
```

```
printf("%c", x);
```

```
printf("%f", (float) x);
// 65.000000
```

- | | |
|---|--|
| <p>① <code>printf("%f", GG);</code> ✓
 Ⓜ</p> <p>② <code>printf("%c", GG);</code> ✓
 ⓑ</p> <p>③ <code>printf("%f", GG);</code> X
 Absurd</p> <p>④ <code>printf("%f", (float)GG);</code>
 GG.000000</p> | <p>⑤ <code>printf("%f", GG.0);</code>
 GG.000000</p> <p>⑥ <code>printf("%d", GG.0);</code> X
 Absurd</p> <p>⑦ <code>printf("%c", GG.0);</code> X
 Absurd</p> <p>⑧ <code>printf("%d", (int)GG.0);</code>
 GG</p> <p>⑨ <code>printf("%c", (char)GG.0);</code>
 ⓑ</p> |
|---|--|