Лекція 2.
Арифметичні операції

План

1) Операнди, операції, вирази
2) Особливості арифметичних операцій
3) Унарні арифметичні операції
4) Зведення типів

Операнди, операції, вирази

Операнд (англ. operand) - це константа, літерал, ідентифікатор, виклик функції, вираз вибору елемента чи більш складний вираз

Операція (англ. *operator*) – спеціальний символ

Комбінація знаків операцій та операндів , результатом якої є конкретне значення, називається виразом.

Вирази можуть бути: *арифметичними, символьними та логічними.*

Арифметичний вираз складається із операндів, арифметичних операцій (+, -, *, /, %), круглих дужок і оператора присвоювання (=).

Приклад.

```
int x, y;
scanf("%d", &x);
scanf("%d", &y);

int z = x + y;
```

Операція «плюс»

операнди

В залежності від кількості операндів, операції поділяються на:

Тип операції	Особливість	Приклади операцій
унарні	один операнд	x = - y; x++;
бінарні	два операнди	z = x + y; r = a * b;
тернарна	три операнди	буде розглянуто пізніше

Розглянемо бінарні арифметичні операції

Позначення	Назва	Приклад
+	додавання	a + b
-	віднімання	a - b
*	множення	a * b
/	ділення	a / b
%	залишок від ділення	a % b

?

$$\frac{a-b}{2c+\dfrac{a}{c+\dfrac{b-5}{c-b}}}$$

(a-b)/(2*c+(a/(c+(b-5)/(c-b))))

Особливості арифметичних операцій

операція ділення для двох цілих чисел завжди дає цілочислельний результат

```
int main()
{
  float x = 1 / 5,
        y = 10 / 3,
        z = 5 / 4;
  printf("x = %f\ny = %f\nz = %f\n", x,
y, z);
  return 0;
}
```

```
x = 0.000000
y = 3.000000
z = 1.000000
```

для отримання дробового результату потрібно, щоб один з операндів був дробовим числом

```
int main()
{
    float x = 1.0 / 5,
          y = 10 / 3.0,
          z = 5.0 / 4.0;
    printf("x = %f\ny = %f\nz = %f\n", x, y,
z);
    return 0;
}
```

```
x = 0.200000
y = 3.333333
z = 1.250000
```

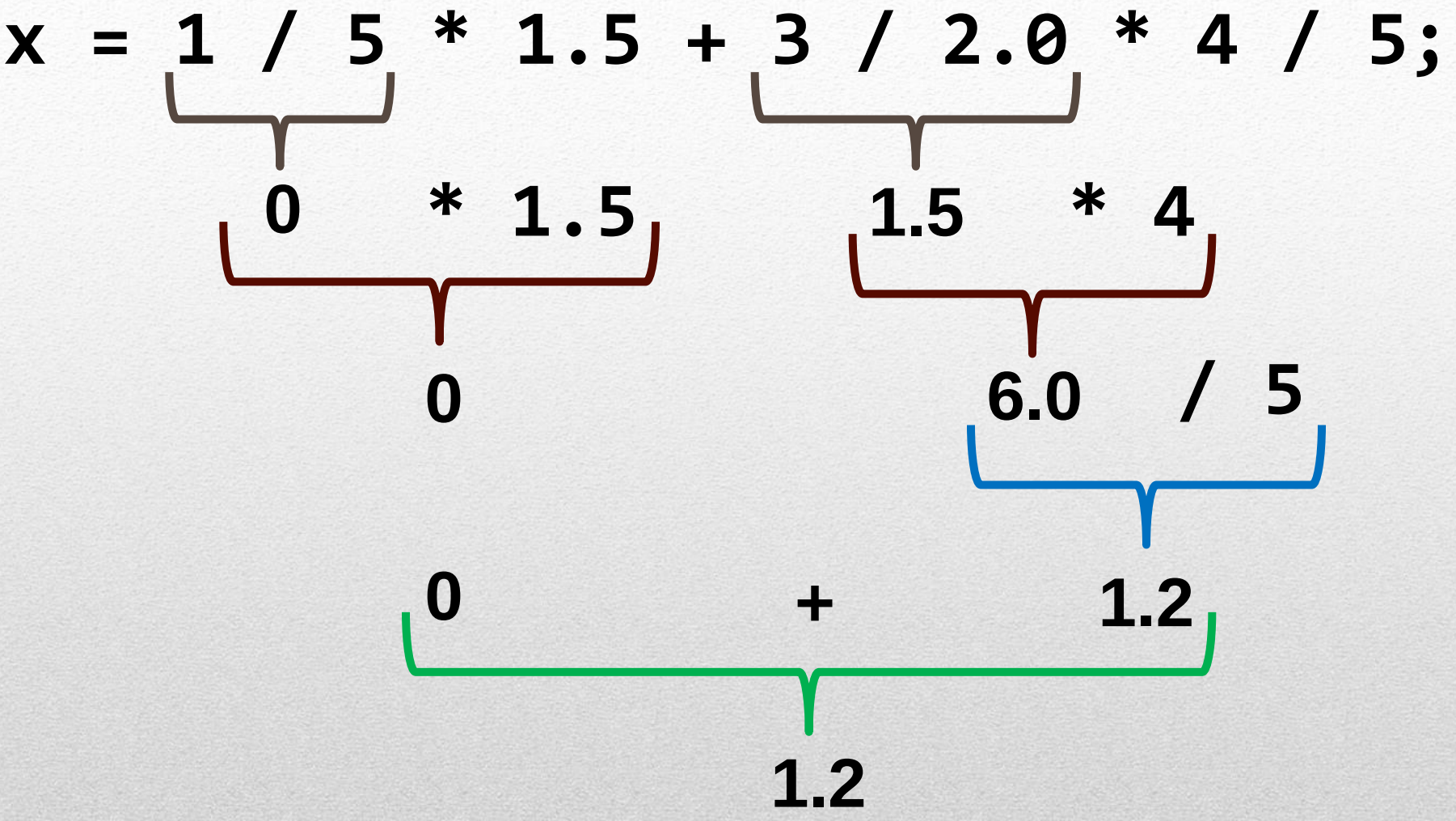
?

Визначте, які значення матимуть змінні x, y та z

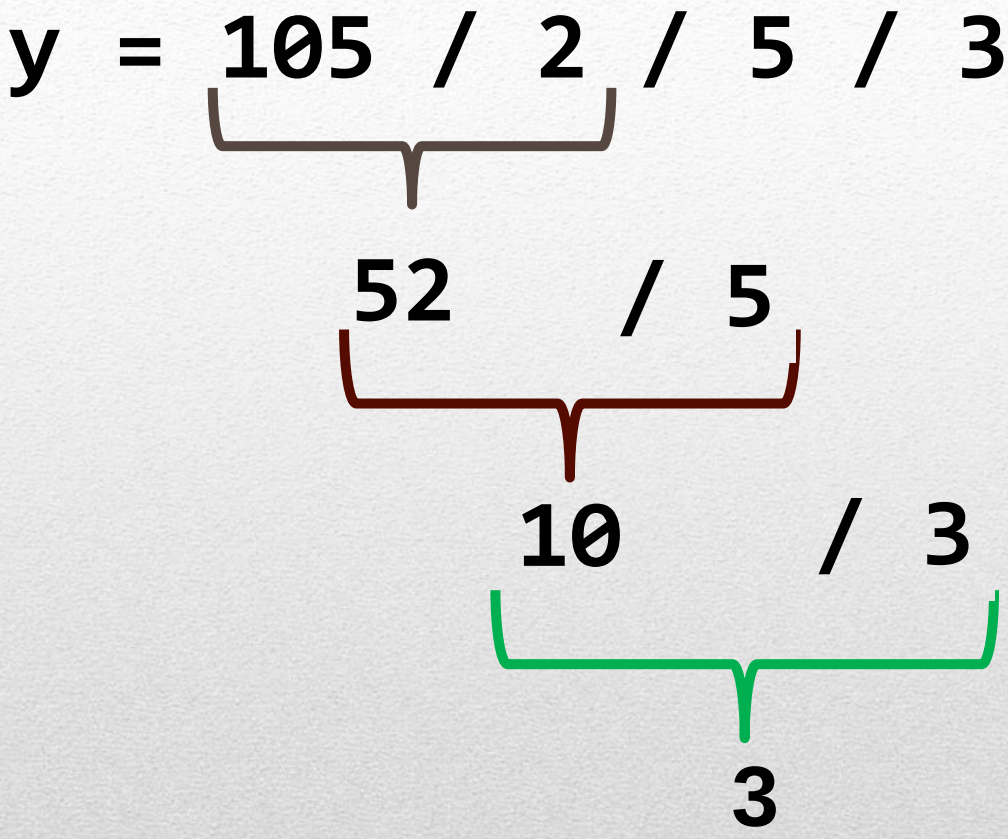
```
int main()
{
  float x = 1 / 5 * 1.5 + 3 / 2.0 * 4 / 5,
        y = 105 / 2 / 5 / 3,
        z = 105 / 2 / 5 / 3.0;

  printf("x = %f\ny= %f\nz= %f\n", x,y,z);

  return 0;
}
```

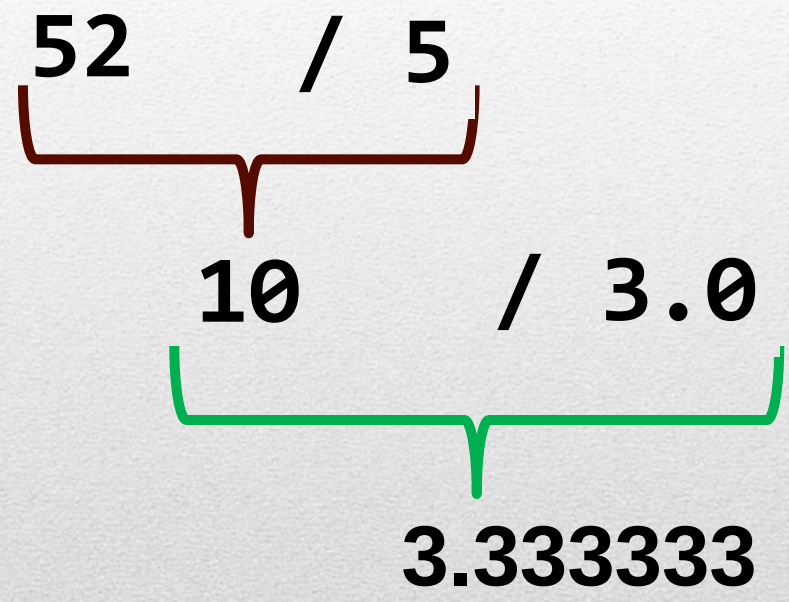


Результат: дробове число **1.2**



Результат: ціле число **3**

```
z = 105 / 2 / 5 / 3.0
```



Результат: дробове число **3.333333**

для зміни порядку виконання операцій використовують круглі дужки

$$y = \frac{x + 5}{a - 3}$$

```
float x, a;
...
y = (x+5)/(a-3);
```

Без круглих дужок вираз виглядав би:

```
y = x+5/a-3;
```

Тоді операції виконувались би в такому порядку:

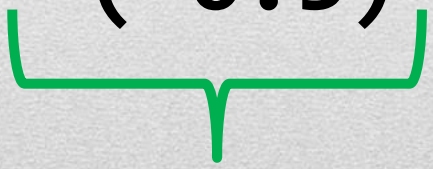
1) ділення; 2) додавання; 3) віднімання

Приклад.

```
float x = -15.5;
float y = -(x + 15);
```

-0.5

-(-0.5)



0.5

Результат: дробове додатне число **0.5**

операція % працює лише для цілих типів даних:

Вираз	Пояснення	Результат
5 % 3	5 / 3 == 1 5 – 1*3 == 2	2
13 % 4	13 / 4 == 3 13 – 3*4 == 1	1
2 % 5	2 / 5 == 0 2 – 0*5 == 2	2

Унарні арифметичні операції

Операції
++ (інкремент) є унарною і додає одиницю до операнда,

-- (декремент) віднімає одиницю від операнда

Розглянемо унарні арифметичні операції

Позначення	Призначення	Приклад
-	зміна знаку	-a
++	інкремент (збільшення на 1)	a++;
--	декремент (зменшення на 1)	a--;
+	нічого не змінює	+a

Коли є присвоювання, спочатку завжди прораховується права частина від знака =

```
x = 5;
```

```
x = x + 1;
```

підставляється значення 5

Тепер вираз виглядає:

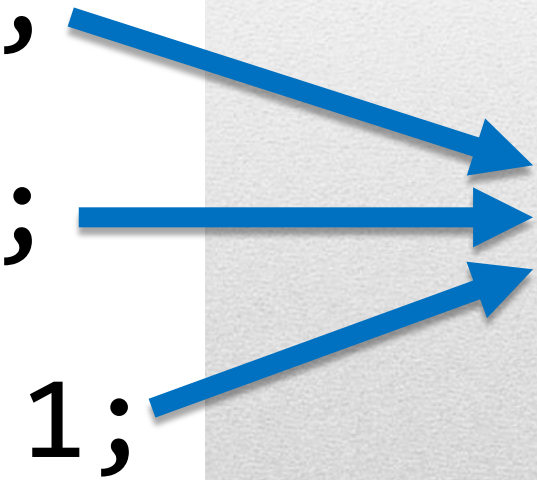
```
x = 5 + 1;
```

Коли права частина прорахована, значення записується у змінну, яка розташована зліва від знака =

```
x = 6;
```

Операція інкременту / декременту може бути записана різними способами:

```
x++;
++x;
x += 1;
```



```
x = x + 1;
```

Інкремент та декремент буває **префіксним** та **постфіксним**

інкремент	`++x`	`x++`
декремент	`--y`	`y--`

префіксний (операція зліва від операнда)

постфіксний (операція справа від операнда)

Особливості:
якщо інкремент чи декремент використовується в середині виразу, то:

- **префіксні виконуються у першу чергу, а потім значення змінних підставляються у вираз;**
- **постфіксні виконуються після підстановки значень змінних у вираз;**

```
int x = 1;
int y = x++ + x++;
```

1 1

Значення виразу: 1 + 1 == 2

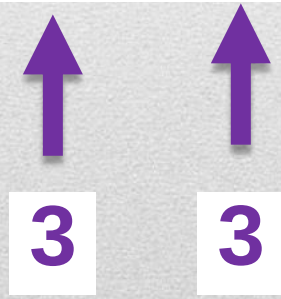
В y записується значення 2

Двічі виконується збільшення x на 1

```
int x = 1;
int y = x + x;
x++; x++;
```

```
int x = 1;
int y = ++x + ++x;
```

```
int x = 1;
x++; x++;          // x = 3;
int y = x + x;
```



Результат: y = 6, x = 3;

```
int x = 1;
int y = x++ + ++x;
```

```
int x = 1;
x++;  // x = 2
int y = x + x;
x++;
```

Результат: y = 4, x = 3

```
int x = 1;
int y = x++ + x++ + ++x;
```

```
int x = 1;
x++;  // x = 2
int y = x + x + x;
x++; x++;
```

Результат: y = 6, x = 4

```
int x = 1;
int y = x + x + ++x;
```

```
int x = 1;
x++;   // x = 2
int y = x + x + x;
```

Результат: y = 6, x = 2

?

Визначте значення кожної змінної після операції, якщо на початку операції всі змінні мають значення рівне 2:

a * = x++

b /= ++x

c = --x + c--

d + =((--x)--)+10?

?

int m=4, n=2;

1) m++ < n --
2) n-- > ++m
3) n ++ < ++ m
4) m-- > ++m

Зведення типів

Перетворення (узгодження) типів виконуються, якщо операнди, які входять до виразу, мають різні типи. Зведення (узгодження) типів здійснюється автоматично за правилом: менш точний тип зводиться до більш точного. Воно буває двох типів:

- Неявне ;
- Явне .

Неявне перетворення типу при арифметичних операціях

типи операндів	тип результату
float / float	float
float / int	float
int / float	float
int / int	int

Зауважимо, що в останньому випадку (і тільки в ньому) здійснюється цілочисельне ділення з відкиданням залишку.

```
int a=2;
float b=3.6;
int c;
Неявне зведення
        c=a*b                   // 3.8*2=7.6
                                //c=7;
Явне зведення
        c=(int) b*a             //3*2=6
        c=(int) (b*a)           //3.8*2=7.6=7
```

?

Якого значення набуде змінна z після виконання наступних дій:

```
float z;
int x=5;
z=x/2;    //   z-?
```

z = (float)(int)x /2=2.0

```
int x = 12;
int y = 7;
double z = x/y;
```

z = (double)((int)x / (int)y)=1.0

```
int x = 12;
int y = 7;
double z = (double)x/y;
```

z = (double)x / (int)y=1.714285714285714