5 Double-precision numbers

In C there are single- and double-precision floating point numbers. Single precision are accurate to about 6 decimal digits and double precision about 15. Single-precision are of no use except where memory is scarce, such as on spaceships.

So we only work with `double` (64 bits).

- `double x,y; // declaration`
- When printing, instead of `%d` use `%f`. This prints correct to 6 decimal places.
- When using `scanf()`, `%f` is ***wrong*** for double precision. This is how to do it:

```
  double z;
  scanf ("%lf", &z);
  printf("z is %f\n", z);
```

- Double-precision numbers which are read using `scanf()` do not need to include a decimal point.

```
  double x,y;
  scanf("%lf %lf", &x, &y);
  printf("%f %f\n", x, y );
a.out
3
4.5
3.000000 4.500000
```

- You can do arithmetic with double precision: $+, -, *, /$. The difference is that division is accurate, not rounded.
- You can mix doubles and integers:

```
  double x; int y;
  x = 1;
  y = 2;
  printf("%f\n", x/y);
```

- You can assign integers to doubles:

```
  int x; double y;
  x = 2;  y = x;
  printf("%d %f\n", x,y);
..... output
2 2.000000
```

- You can assign doubles to integers. The result is rounded towards zero.

```
double x; int y;
x = -2.5;
y = x;
printf("%f %d\n", x,y);
```