

Arrays

Single Dimensional Arrays

```
double[] score;
score = new double[150];

score[index] = 1.50;
score[10] = 3.00;
score[35] = score[45] + score[index];

int test = score.length;

boolean[] flags = new boolean[20];
```

150 ang size ng array; ang value ng **test** ay 150.

zero hanggang 19 ang index ng **flags**

Tandaan:

- Square brackets [] ang ginagamit.
- Kailangan mag-**new** bago gamitin ang array. Kailangan sabihin kung gaano kalaki ang array dito at hindi na mapapalitan ito. Kung ano ang sinulat dito ganoon na 'yun kalaki. Kung kailangan nyo palitan (palakihin) ang array, panibagong **new**; pero mabubura lahat ng nasa lumang array.
- Ang **index** nag-u-umpisa sa zero hindi 1 at ng maximum index ay (*size* - 1).
- **Integer** o **char** lang ang pwedeng gamitin sa index.
- **Integer** lang ang pwedeng gamitin sa size kapag nag-**new** kayo. Pwedeng variable, pwedeng constant, basta't integer.
- Positive (non-zero) integer lang ang pwedeng gamitin para sa size at index.
- Para malaman kung ilan ang laman ng array, gamitin ang **length**.
- lisang type lang ang laman ng isang array; hindi pwedeng iba't-ibang type sa loob ng isang array.

Two-Dimensional Arrays

```
int[][] grid;
grid = new int[10][5];

int uno = grid.length;
int dos = grid[0].length;
int tres = grid[3].length;
```

uno = 10

dos = 5

tres = 5

Tandaan:

- Katulad ng single-dimensional array, hindi pwedeng magbago ang laki.
- Pwedeng iba't-ibang laki ang mga array sa loob ng array; pero para sa atin (CS4) hindi natin ito gagamitin. Para sa atin, parepareho ang laki ng array sa loob.

At iba pa

- may ilang short cut sa pagsusulat ng code

long way	shortcut
c = c + 1	c++
z = z - 1	z--
d = d + 10	d += 10
q = q - 8	q -= 8
s = s * 5	s *= 5
t = t / 3	t /= 3

- kung kinakailangan lumabas kaagad sa isang **for loop** o **while loop**, gamitin ang **break**

```
while(true)
{
    int hula = Console.random(1,100);
    if (hula > 95)
    {
        Console.println("kalayaan!");
        break;
    }
    Console.println("bihag pa ako...");
}
Console.println("nasa labas na po...");
```

oras na **hula > 95**,
lalabas agad ito sa
(infinite) while loop

dito ang bagsak
pagkatapos ng **break**

- kung kinakailangan bumalik sa umpisa ng loop na hindi dadaanan ang ibang statements, gamitin ang **continue**

halimbawa (while loop):

```
int ans = -1;
while (ans < 1)
{
    Console.println("Enter a positive integer: ");
    ans = Console.readInteger();

    if (ans < 1)
    {
        Console.println("positive integers lang po");
        continue;
    }
}

// do more stuff
Console.println("do more stuff");
```

dito ang bagsak pagkatapos ng **continue**

halimbawa (for loop):

```
int marked = 0;
int i;
for(i = 0; i < 1000; i = i + 1)
{
    String s = Console.readLine();
    sample[i] = s;

    if ((i / 5) == 0)
    {
        // no marks every 5th item
        continue;
    }

    sample[i] = "MARK:" + sample[i];
    marked = marked + 1;
}
```

dadaan muna sa **i=i+1** bago tingnan ang *expression* (i<100) **pagkatapos ng continue**

hindi gagawin ang dalawang lines na ito kapag nag-**continue**