

# Workshop I

---

*Composing with Algorithms*  
<http://www.bjarni-gunnarsson.net>

# Format

---

10 tasks are described that need implementation with patterns and simple synthesis.

Students shall work in groups to solve at least some of the tasks during class.

The final 30 minutes of class should involve a short presentation of each group where they show or discuss what they have done.

*Tasks*

*// For today best use the simple sine synth or the default one.*

```
SynthDef(\sine, {  
  |amp=0.1, freq=440, pan=0|  
  var env, sig;  
  env = EnvGen.kr(Env.perc, doneAction:2);  
  sig = SinOsc.ar(freq) * env;  
  sig = Pan2.ar(sig, pan, amp);  
  Out.ar(0, sig);  
}).add;
```

*// Then use Pbind to bind the synth to compositional patterns. The parameters (\freq and \amp) should match the ones of the Synth. \dur is an exception since it concerns the duration of an event.*

```
Pbind(  
  \instrument, \sine,  
  \freq, Pseq([100, 800, 600], inf), // 3 frequencies are repeated  
  \amp, Env([0.0, 1.0, 0.0], [3, 4]), // Volume goes from 0 to 1 and to 0 again  
  \dur, Pwhite(0.05, 0.1) // Each note has a random duration from 50 to 100 ms  
)
```

`.play`

# Tasks I

---

1. Implement a process where a synth is played first very fast and then very slowly. The duration pattern should alternate between the two types of duration.

*(hint: see Pseq with nested patterns)*

2. Implement a process that plays two synths interchangeably where the probability of synth a is 25% and event b is 75%

*(hint: see Pwrand for weighted randomness)*

3. Implement a stochastic process where low pitches have a longer duration than high ones.

*(hint: see Pkey to couple parameters)*

## Tasks 2

---

4. Implement a Process based on two or more layers where moments occur with all layers playing at the same time and others with only a single layer playing.

*(hint: see Ppar for parallel patterns).*

5. Implement a sequence where pitches are chosen randomly. The randomness should be very wide in range in the beginning but decrease and settle once it continues.

*(hint: see Penv or simply Env for gradual shapes).*

6. Implement a process with various layers where each layer appears and later disappears gradually.

*(hint: see Penv or simply Env for gradual shapes).*

## Tasks 3

---

7. Implement a process that oscillates between random values and repeated ones.

*(hint: see `Pstutter` for repeating values).*

8. Implement a sequence of two layers, both using brownian motion for pitch and duration values. One layer should stop before the other.

*(hint: see `Pbrown` for brownian motion).*

9. Implement a process with at least three layers where moments occur where all are playing but also where each one plays individually.

*(hint: see `\rest` for rest values).*



## Tasks 4

---

10. Implement a process where pitch values are determined either according to cauchy or exponential distribution. Dynamics should be determined with a geometric rise.

*(hint: see  $P_{cauchy}$ ,  $P_{exprand}$  and  $P_{geom}$ ).*

