

Announcements — Tues., Jan. 18

- For next time, read *Learning Computing with Robots*, ch. 1. **This is a change from what I told you last time!**
- The Blackboard site is up and operational!
 - not much content
 - mostly links back to my course homepage
- Printed copies will probably be available from Graphic Creations
 - 1809 Lake Avenue, Knoxville
 - *How to Think Like a Computer Scientist*: \$20
 - *Learning Computing with Robots*: \$30
 - I don't know if they print copies in advance; you might want to call them (522-6221)
- I will also make them available through lulu.com. TCS is currently available for \$13.28 + S/H. See course home page.



The Algorithm

Slides (with modifications) from
Institute for Personal Robots in Education (IPRE)

2011-01-18

2

CS 102 / ECE 206 with Robots

The Algorithm

“Algorithm:

n. A finite set of unambiguous instructions performed in a prescribed sequence to achieve a goal” — *The American Heritage Science Dictionary*

The Algorithm

Al-Khwarizmi was a Persian mathematician who wrote a book on calculating with Hindu numerals in the 9th century CE. When translated to Latin, a pluralized form of his name (*algorismus*) became synonymous with a system of calculation.



Pizza Dough Recipe (Algorithm)

1. Gather Ingredients
2. Combine sugar (1tbs), salt (1tbs), olive oil (1tbs), flour (1c) in mixing bowl
3. Turn on mixer
4. Add 1/4 cup of flour
5. If dough comes off the sides, go to step 6, otherwise go back to step 4
6. Knead 15 minutes
7. Let rest for at least 45 minutes in warm area



1 dough ball

***N*-Pizza Dough Algorithm**

1. Gather Ingredients
2. Combine sugar (N tbs), salt (N tbs), olive oil (N tbs), flour (N c) in mixing bowl
3. Turn on mixer
4. Add $N/4$ cup of flour
5. If dough comes off the sides, go to step 6, otherwise go back to step 4
6. Knead 15 minutes
7. Let rest for at least 45 minutes in warm area



***N* dough balls**

***N*-Pizza Dough Algorithm**

1. Gather Ingredients
2. Combine sugar (N tbs), salt (N tbs), olive oil (N tbs), flour (N c) in mixing bowl
3. Turn on mixer
4. Add $N/4$ cup of flour
5. If dough comes off the sides go to step 6, otherwise go back to step 4
6. Knead 15 minutes
7. Let rest for at least 45 minutes in warm area

Sequence of
Statements

***N* dough balls**

***N*-Pizza Dough Algorithm**

1. Gather Ingredients

2. Combine sugar (N tbs), salt (N tbs), olive oil (N tbs), flour (N c) in mixing bowl

Variable

3. Turn on mixer

4. Add $N/4$ cup of flour

5. If dough comes off the sides go to step 6, otherwise go back to step 4

6. Knead 15 minutes

7. Let rest for at least 45 minutes in warm area

N dough balls

N-Pizza Dough Algorithm

1. Gather Ingredients
2. Combine sugar (N tbs), salt (N tbs), olive oil (N tbs), flour (N c) in mixing bowl
3. Turn on mixer

4. Add $N/4$ cup of flour

5. If dough comes off the sides go to step 6, otherwise go back to step 4

Conditional

```
graph LR; A[5. If dough comes off the sides go to step 6, otherwise go back to step 4] --> B[Conditional]
```

6. Knead 15 minutes

7. Let rest for at least 45 minutes in warm area

N-Pizza Dough Algorithm

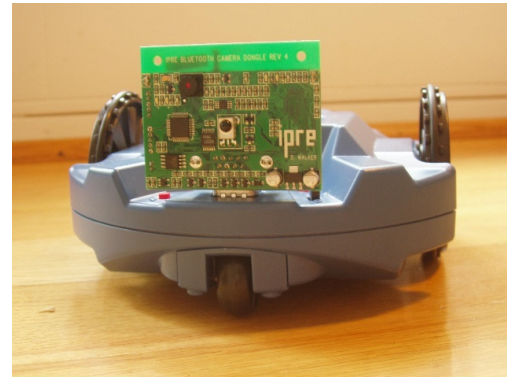
1. Gather Ingredients
2. Combine sugar (N tbs), salt (N tbs), olive oil (N tbs), flour (N c) in mixing bowl
3. Turn on mixer
4. Add N/4 cup of flour
5. If dough comes off the sides go to step 6, otherwise go back to step 4
6. Knead 15 minutes
7. Let rest for at least 45 minutes in warm area

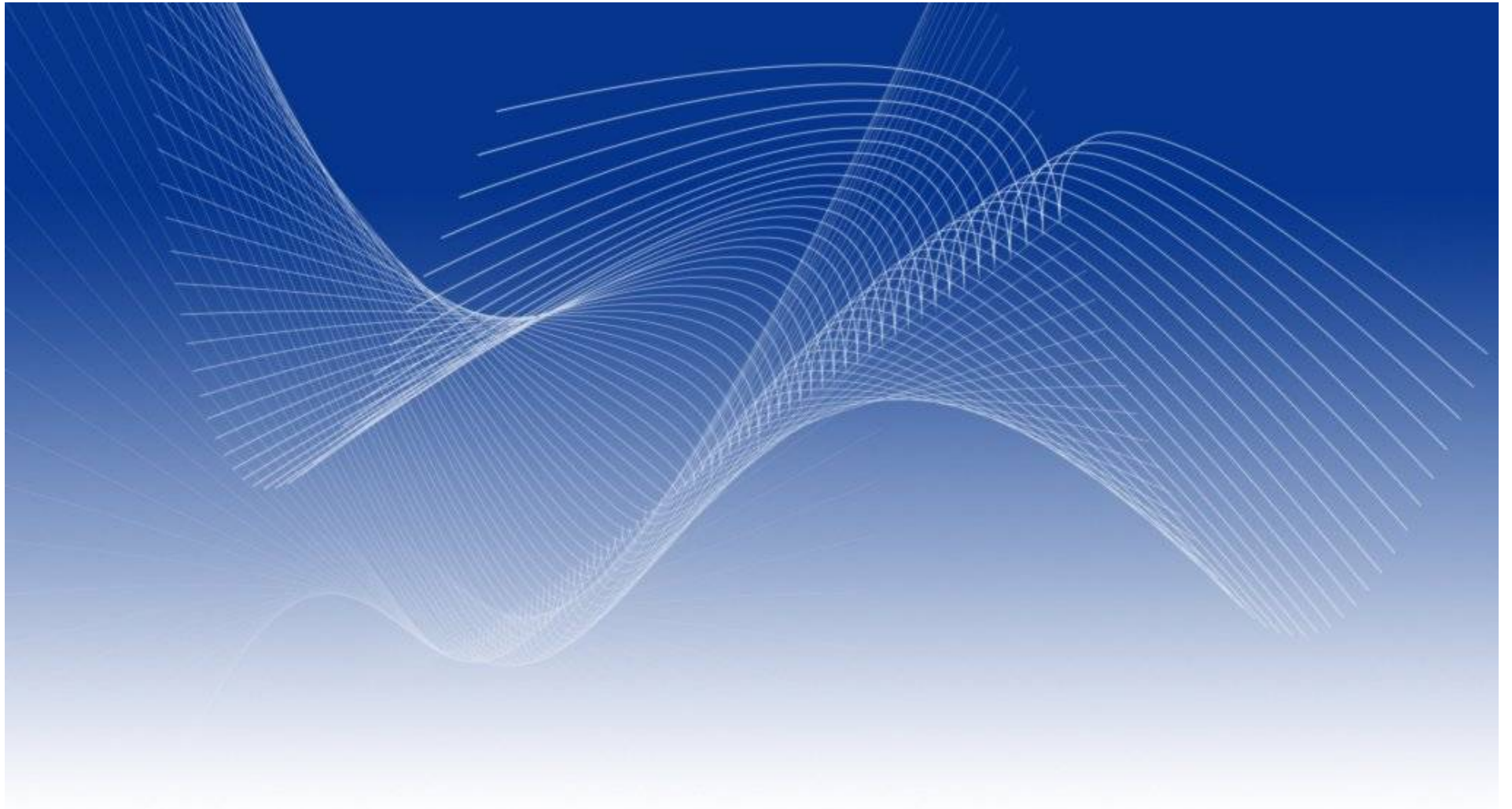
```
graph LR; S3[3. Turn on mixer] --- J(( )); S6[6. Knead 15 minutes] --- J; J --> SA[Subroutines Mini-algorithms];
```

Subroutines
Mini-algorithms

Programs — Algorithms at Work

- Programs are the way we communicate with a computer
- Specify the algorithm
- This class uses the C++ programming language
- Rather than pizza dough, we will use a robot as our favorite example





The Way of the Program

12

2011-01-18

CS 102 / ECE 206 with Robots

Programming languages

- There are thousands of programming languages!
- High-level languages:
 - These are some languages I have programmed in: A, Ada, Algol 60, Algol 68, APL, Argot, BASIC, BCPL, Bliss, Chrysalis, COBOL, C++, FORTRAN, FP, GPL, LISP, Mathematica, NetLogo, Pascal, ϕ , PL/M, Prolog, Python, Scheme, Simula 67, Smalltalk, StarLogo, Ω .
 - These are some well-known and popular languages that I have *not* programmed in: C, C#, Forth, Haskel, Java, MatLab, Miranda, Pearl, PL/I.
- The point? *In your career you will have to learn and use many HLLs!*
- The key skill is *how to program*
- You should also be able to pick up new HLLs quickly
- There are also low-level languages

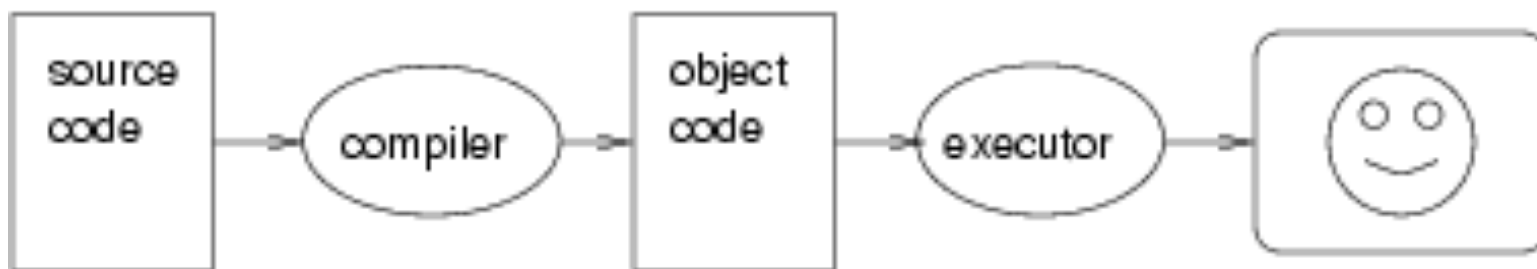
Interpreters



The interpreter
reads the
source code...

... and the result
appears on
the screen.

Compiler



The compiler reads the source code...

... and generates object code.

You execute the program (one way or another)...

... and the result appears on the screen.

What is a program?

- A program is a sequence of instructions for performing a computation
- That is, a precise description of an algorithm
- Instructions include:
 - input
 - output
 - math
 - testing
 - repetition
- Programs need to be comprehensible to people as well as computers!

Debugging

- Compile-time errors
- Run-time errors
- Logic errors

- Experimental debugging (hypothesis testing)

Formal vs. natural languages

- Natural languages
 - ambiguity
 - redundancy
 - literalness
- Formal languages
- Programming languages are formal languages for expressing algorithms

A first program — “Hello World”

```
#include <iostream>
using namespace std;

// main: generate some simple output

int main ()
{
    cout << "Hello, world." << endl;
    return 0;
}
```