

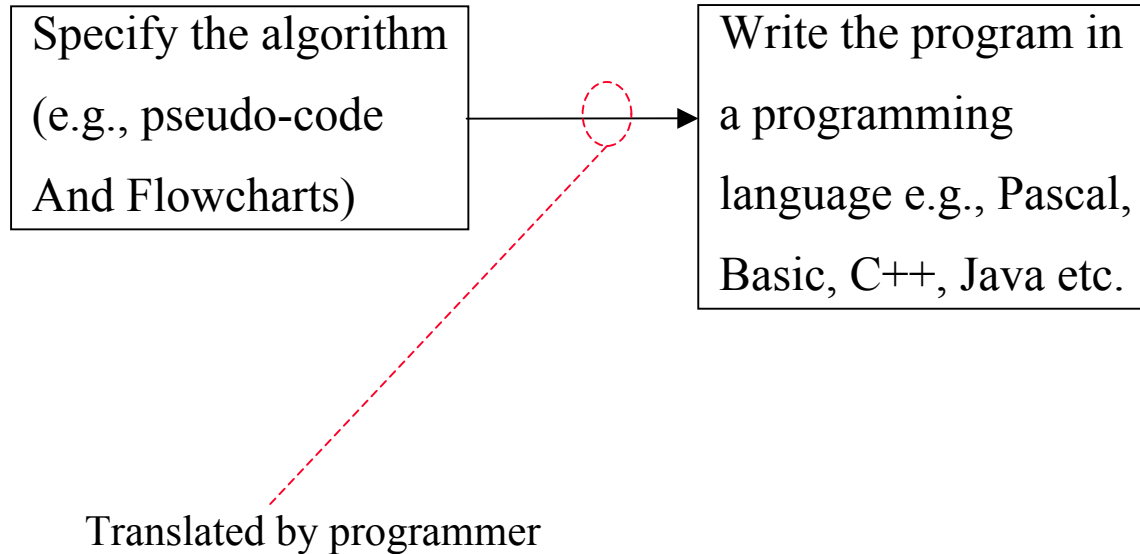
# **Creating Actual Computer Programs**

**This section of notes describes how a computer program written by a person is translated to a form that can be understood by a computer. Also you learn about the basic structure of Pascal programs.**

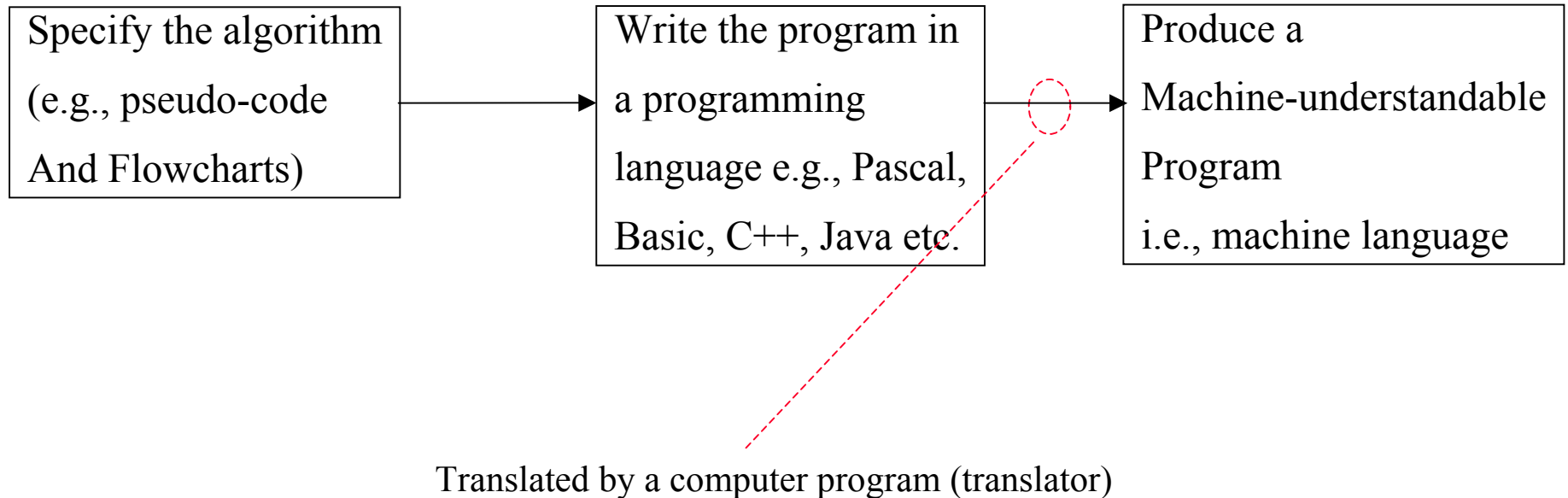
# Creating Computer Programs: What You Know

Specify the algorithm  
(pseudo-code and  
Flowcharts)

# Creating Computer Programs: What You Will Learn



# Creating Computer Programs: What You Will Learn (2)



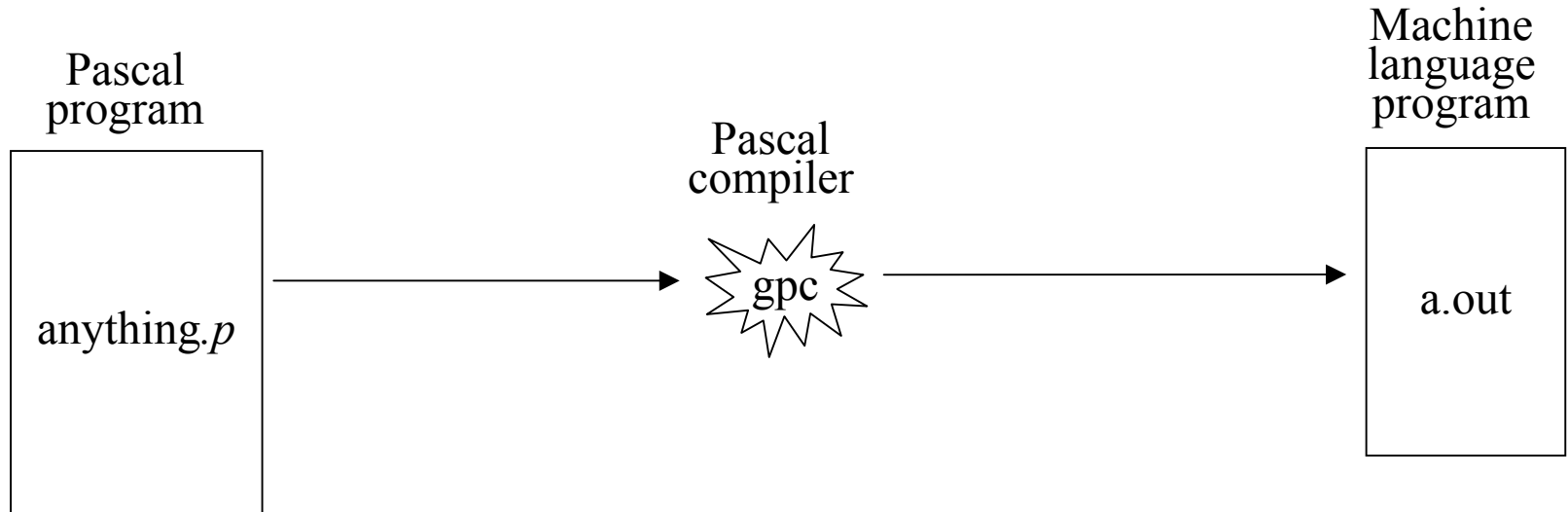
# Translators

Convert computer programs to machine language

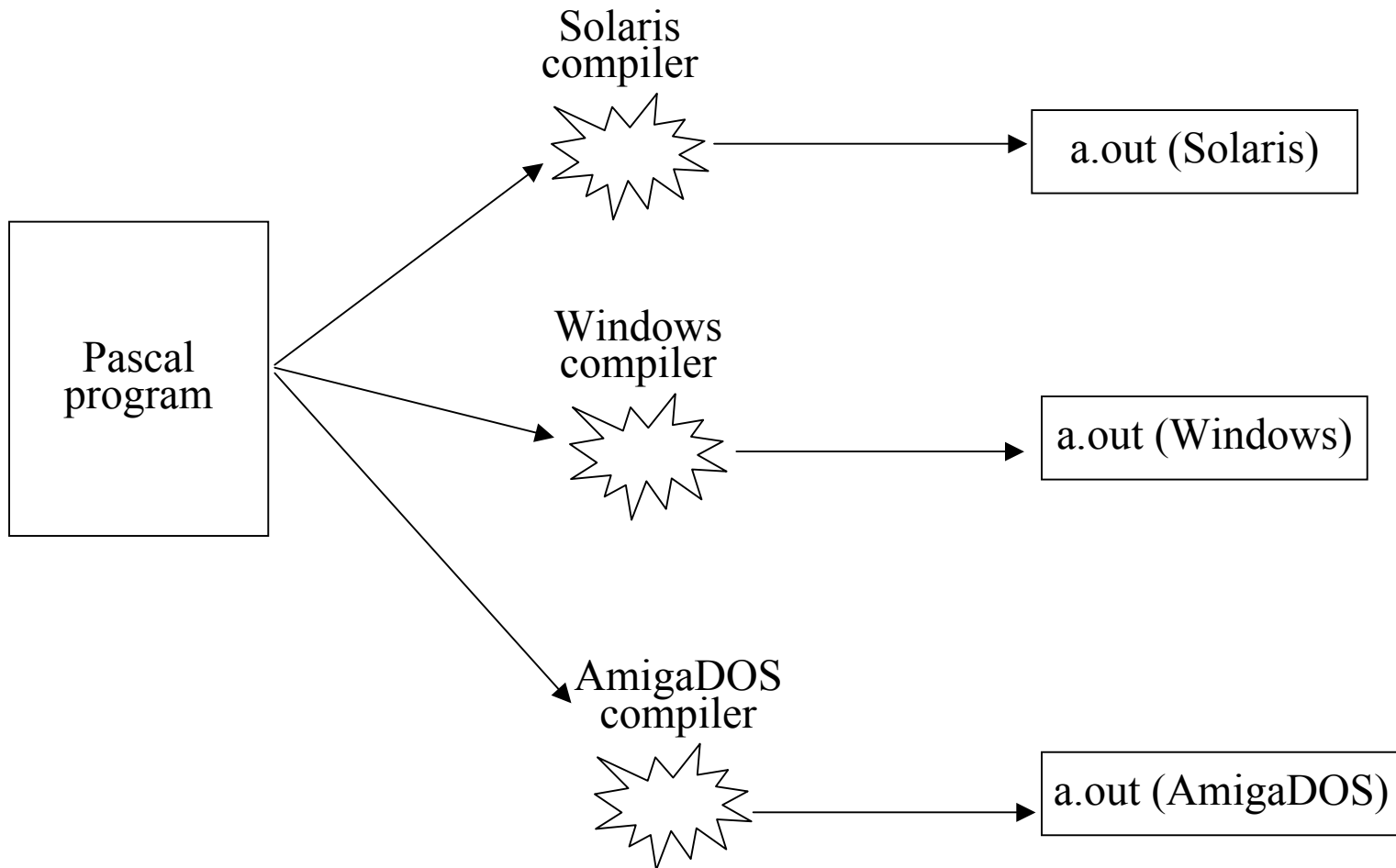
Types

- 1) Interpreters
- 2) Compilers

# Compiling Programs On The CPSC Network: Basic View



# Compiling Programs On Different Operating Systems



# Components Of Computer Programs

## Headers

- Program name, version number, date last modified, what program does
- One form of documentation – comments for the reader of the program (not the computer):
  - (\* Marks the beginning of the documentation
  - \*) Marks the end of the documentation

## Declarations

- List of variables

## Statements

- The instructions in the program that actually gets stuff done
- Each statement is separated by a semicolon ";"

Reminder: The filename for Pascal program must end with dot-p, ".p".



# The Smallest Pascal Program

(\*  
The smallest compilable Pascal program written by James Tam

\*)  
program smallest  
begin  
end.

Note: The name "smallest" should match the filename "smallest.p". You can find an online version of this program in the Unix file system under `/home/231/examples/intro/smallest.p`

# Summary

How is a computer program translated into a machine-understandable form

What is the basic structure of a Pascal program

What are the fundamental parts of Pascal programs