

### Another Category/Partition Example

- **Program:** Java Compiler
- **Syntax:** `javac program.java ...`
- **Function:** The **javac** tool reads class and interface definitions, written in the Java programming language, and compiles them into bytecode class files. There are two ways to pass source code file names to **javac**:...

### ITFs

1. Correctly identifies syntactic errors
2. Correctly identifies environment issues (e.g. missing Classes, types etc)
3. Correctly translates a correct java program into a class file
4. Correctly interprets command line options
5. Correctly applies optimisations
6. ...

### ITF: Correctly identifies syntax errors

- First we identify the parameters and environment conditions that affect the behaviour of the system. For the full **javac** program there are more but we restrict here to illustrate the idea while keeping the presentation to tolerable length:
  - Parameter: java filename
  - Parameter: command line options
  - Environment: File system
  - Environment: Various shell variables
  - Environment: Libraries and other external code

### Finding Categories

- Quoting Ostrand and Balcer:
  - *“A category is a major property or characteristic of a parameter or environment condition. Selecting the categories is done by careful reading of the specification. For each parameter or environment condition the tester marks phrases in the specification that describe how the functional unit behaves with respect to some characteristic of the parameter or environment condition. Each characteristic that can be identified in this way is recorded as a category.”*

## Categories for Javac

- Parameter: java filename
  - Valid java filename (e.g. x.java)
  - Invalid filename (e.g. x.py)
- Parameter: command line options
  - Specifies source version
  - Warning messages
- Environment: File system
  - Source file not present
  - Source file parsing errors
  - Source file type errors
  - Source file naming errors
  - classpath directories absent
  - classpath directories present
- Environment: Various shell variables
  - Invalid classpath
  - Valid classpath
- Environment: Libraries and other external code...

## Choices

- Parameter: java filename
  - Valid java filename (e.g. x.java) simple name, compound name
  - Invalid filename (e.g. x.py) simple name, compound name
- Parameter: command line options
  - Specifies source version: less than 1.5, 1.5 or greater
  - Warning messages: option present, option absent
- Environment: File system
  - Source file - not present: no file
  - Source file - parsing errors: no errors, one error, multiple errors
  - Source file - type errors: no errors, one error, multiple errors
  - Source file - naming errors: no errors, one error, multiple errors
  - classpath directories absent: no missing classes, missing classes
  - classpath directories present: no missing classes, missing classes
- Environment: Various shell variables
  - Invalid classpath: invalid syntax
  - Valid classpath: no paths, single path, multiple path
- Environment: Libraries and other external code...

## Number of Test Specifications

- 2.2.2.2.1.3.3.3.2.2.1.3 = 5184
- But many of these are not really valid combinations, for example:
  - An invalid file name only needs to be checked once or twice
  - If a valid filename is not present in the file system we only need one or two tests not many.
  - Possibly it is not worth looking at type or naming errors if we have syntax errors

## Adding Conditions

- Parameter: java filename
  - Valid java filename (e.g. x.java) simple name, compound name
  - Invalid filename (e.g. x.py) simple name, compound name
- Parameter: command line options
  - Specifies source version: less than 1.5, 1.5 or greater
  - Warning messages: option present, option absent
- Environment: File system
  - Source file - not present: no file
  - Source file - parsing errors: no errors [property OkSyntax], one error, multiple errors
  - Source file - type errors: no errors, one error[if OKSyntax], multiple errors[if OKSyntax]
  - Source file - naming errors: no errors, one error[if OKSyntax], multiple errors[if OKSyntax]
  - classpath directories absent: no missing classes, missing classes
  - classpath directories present: no missing classes, missing classes
- Environment: Various shell variables
  - Invalid classpath: invalid syntax
  - Valid classpath: no paths, single path, multiple path

## Add Conditions

- Can you think of more conditions to add?
- Try introducing all those you can justify.
- Once you have done that calculate the number of test specification you generate.