1. Enum

1.1 Enum syntax

```
Enum ENUM_NAME {
   (enum_stmt) +
}
enum_stmt =
   ENUMERATOR_NAME (=INITIALIZER),
```

**Enum**: Start of an enum block.

**ENUM_NAME**: Required name of the enum block. This name applies scope to the enumerator elements within the enum block.

**ENUMERATOR_NAME**: Required name of the enumerator element. Each enumerator name within the enum block must be unique.

**INITIALIZER**: Optional assignment value for the enumerator. If the initializer is omitted, the enumerator takes on the value of the previous enumerator + 1. The first enumerator, if not assigned with an initializer, is assigned the starting value of 0. Each enumerator value within the Enum block must be unique, and must be assigned a positive integer value.

```
Enum FailMode {
   # Jim’s addition — what value would we want to use for
   # NOT_EXECUTED? Per above, values must be > 0; do we want to
   # allow =0 or <0 as well? Or assign a fairly high integer > 0?
   NOT_EXECUTED = -1,  #?? What value for NOT_EXECUTED?
   PASS = 0,  // 0
   EXCEPTION_SOFT = 1,  // 1 Software exception, e.g., divide by 0
   EXCEPTION_HARD = 2,  // 2 Hardware exception
   FAIL_UNITS = 3,  // 3 Units mismatch
   FAIL_FNC = 4,  // 4 Functional or functional part of parametric
   FAIL_PRM = 5,  // 5 Parametric, e.g., search endpoints
   FAIL_BOTHLIM = 6,  // 6 Lower and upper limit
   FAIL_HILIM = 7,  // 7 Upper limit
   FAIL_LOLIM = 8,  // 8 Lower limit
   INDETERMINATE = 9,  // 9 Limits compared to result None
}
```