12.13 Random weighted case - randcase

```
statement_item ::=  ...                                                          // from Annex ???
                    { attribute_instance } randcase  randcase_item { randcase_item } endcase

randcase_item ::=  expression : statement_or_null
```

The keyword **randcase** introduces a case statement that randomly selects one of its branches. The randcase item expressions are non-negative integral values that constitute the branch weights. An item's weight divided by the sum of all weights gives the probability of taking that branch. For example:

```
randcase
   3 : x = 1;
   1 : x = 2;
   4 : x = 3;
endcase
```

The sum of all weights is 8, so the probability of taking the first branch is 0.375, the probability of taking the second is 0.125, and the probability of taking the third is 0.5.

If a branch specifies a zero weight then that branch is not taken. If all **randcase** items specify zero weights then no branch is taken and a warning may be issued.

The **randcase** weights can be arbitrary expressions, not just constants. For example:

```
byte a, b;

randcase
   a + b   : x = 1;
   a - b   : x = 2;
   a ^ ~b  : x = 3;
   12'b800 : x = 4;
endcase
```

The precision of each weight expression is self-determined. The sum of the weights is computed using standard addition semantics (maximum precision of all weights), where each summand is unsigned. Each weight expression is evaluated at most once (implementations may cache identical expressions) in an unspecified order. In the example above, the first three weight expressions are computed using 8-bit precision, the fourth expression is computed using 12-bit precision; the resulting weights are added as unsigned values using 12-bit precision. The weight selection then uses unsigned 12-bit comparison.
Each call to **randcase** retrieves one random number in the range zero to the sum of the weights. The weights are then selected in declaration order: smaller random numbers correspond to the first (top) weight statements.

Randcase statements exhibit thread stability. The random numbers are obtained from $urandom_range, thus, random values drawn are independent of thread execution order.