Declaration Qualifiers
Type Qualifier

Last updated 7/25/25

Declaration Qualifiers – Type Qualifier

- Declaration – Full Structure
  - This applies to variables and functions

  - Declarations consist of 4 elements
    - Storage Class
    - Type Qualifier
    - Type Specifier
    - Declarator

  - Format
    - (Storage_Class) (Type_Qualifier) Type_Specifier Declarator;
      int foo;
      - () indicates optional

Declaration Qualifiers – Type Qualifier

- Declaration – Type Qualifier
  - Provides additional information on how a variable may be accessed or modified
  - Used by the compiler to allow or prevent various optimizations
  - 4 type qualifiers
    - const
    - volatile
    - restrict
    - _Atomic

Declaration Qualifiers – Type Qualifier

- const
  - The variable cannot be changed after it is initialized
  - Frequently used with arrays and functions
    - Since the array is passed by reference, identifying it as const tells the compiler is should not be changed by the function

```
void ary_copy(const int src[], int dest[]);
```

- volatile
  - The variable may be changed by someone other than the current code
  - Prevents the compiler from optimizing the variable away if never written to
  - Common in embedded systems where hardware registers may change variable values

```
volatile int adc_reg_0;
```

Declaration Qualifiers – Type Qualifier

- restrict
  - Indicates there is only one way to reference a pointer
  - Allows additional compiler optimization

```
void ary_fn(restrict int ary1[], restrict int ary2[]);
```

- _Atomic
  - Indicates changes to the variable must be made without interruption from other processes
  - Primarily used with concurrent programming (threaded)

```
_Atomic int shared_reg_0 = 0;
```