The `const` modifier

- **The `const` keyword**

  - `const`:

    - The **keyword `const`** is used to **specify** that a **variable *cannot*** be **modified**:

      Example:

      ```
      int  a ;            // a is an int variable

      int const a ;       // a is an int variable that cannot be modified
      ```

      (It **actually** says that: the **expression** `a` ***cannot* be modified**)

    - A **variable** defined with the `const` **keyword** can ***only*** be given an ***initial* value**

    Note:

    - This **feature** is a **language construct *built* into** the **C compiler**
    - In other words:
      - When the **C compiler** detects that the program uses an **assignment statement** with a `const` **modified variable**, the **C compiler** will **report an error**.

  - **Example:**

    ```
    int main(int argc, char *argv[])
    {
       int const a = 4;    // Initial value

       printf (" a = %d\n", a);

       a = 1;          // Illegal !!!

       printf (" a = %d\n", a);
    }
    ```

    Result:

    ```
    UNIX>> gcc const1.c
    const1.c: In function 'main':
    const1.c:12: error: assignment of read-only variable 'a'
    ```

  - **Example Program:** (Demo above code)   Example

- Prog file: click here

How to run the program:

- **Right click** on link(s) and **save** in a scratch directory
- To compile: `gcc const1.c`

  Check out the compile error !

- **Postscript: alternate syntax**
  - **Alternate syntax** to define a ***inmutable* variable**:

```
const  DataType  varName ;
```

Example:

```
const  int  a ;
```

- **Note:**

  - This **syntax** is **preferred**:

```
DataType  const  varName ;
```

This will be explained later in this webpage: **click here**