Software Engineering (CS401)

Lab Assignment 2

**U19CS012**

Q1.) Write a **C program** having some **global variables** that are **declared** but **not used** anywhere in the code. Run Splint for this C code and report the error generated.

**Code**

*#include* <stdio.h>

*// Unused variables*

*// Global Variable [Uninitialized]*

int global\_var\_1;

*// Global Variable [Initialized]*

int global\_var\_2 = 10;

int main()

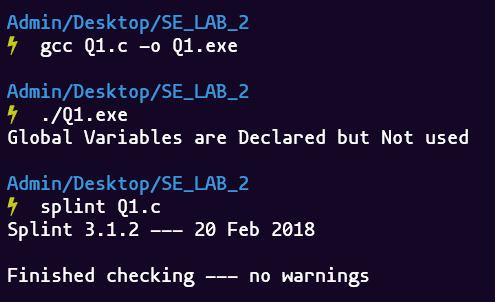
{

    printf("Global Variables are Declared but Not used\n");

*return* 0;

}

**Output**



**No Error** Generated by Splint Tool.

Q2.) Write a **C program** having some **global variables** that are **declared** but **not initialized**. Return this **uninitialized variable** in the main function. Run Splint for this C code and report the error generated.

**Code**

*#include* <stdio.h>

*// Unused variables*

*// Global Variables [Uninitialized]*

int global\_var\_1, global\_var\_2;

int main()

{

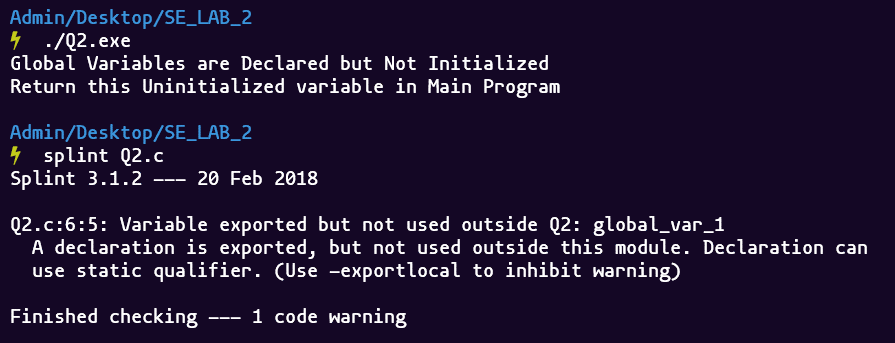
    printf("Global Variables are Declared but Not Initialized\n");

    printf("Return this Uninitialized variable in Main Program\n");

*return* global\_var\_1;

}

**Output**



Q3.) Write a **C program** having some **global variables** that are **declared** but **not initialized**. Initialize some local variable using this **uninitialized global variable**. Run Splint for this C code and report the error generated. (For instance, assume global variable ‘a’ is declared as ‘int’ in the code. In the main function you can perform some operation such as ‘int b =a’. This code should generate some error as the variable ‘a’ is not initialized in the code.)

**Code**

*#include* <stdio.h>

*// Unused variables*

*// Global Variables [Uninitialized]*

int global\_var\_1, global\_var\_2;

int main()

{

    printf("Global Variables are Declared but Not Initialized\n");

    printf("Initialize the Local Variables with Uninitialized Global Variables\n");

    int local\_var\_1;

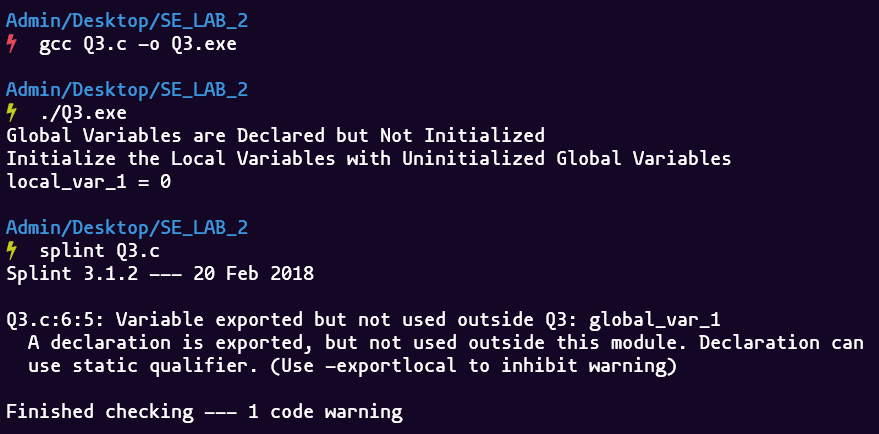
    local\_var\_1 = global\_var\_1;

    printf("local\_var\_1 = %d\n", local\_var\_1);

*return* 0;

}

**Output**



Q.4) Write a **C program** having **structure** as **global variable**. This structure can have more than two fields. Except one field, you can **initialize values** to all fields in the structure. Run Splint for this C code and report the error generated. (This code should generate error as you have one uninitialized field in structure)

**Code**

*#include* <stdio.h>

*// Global Structure*

struct student

{

    char \*name;

    int age;

    float per;

};

float get\_percent()

{

*// Local Structure*

    struct mark

    {

        float m1, m2, m3;

    } s;

    s.m1 = 80.5;

    s.m2 = 84.5;

    s.m3 = 90;

    printf("Physics     : %f\n", s.m1);

    printf("Chemistry   : %f\n", s.m2);

    printf("Mathematics : %f\n", s.m3);

    float percent = (s.m1 + s.m2 + s.m3) / 3;

*return* percent;

}

int main()

{

    printf("Structure as Global Variable\n");

    printf("Except one field, you can initialize values to all fields in the structure.\n");

    struct student o =

        {

            .name = "Raju",

            .age = 20,

        };

    o.per = get\_percent();

    printf("\nName    : %s", o.name);

    printf("\nAge     : %d", o.age);

    printf("\nPercent : %f", o.per);

*return* 0;

}

**Output**

