

EXPERIMENT 11

- ASHISH KUMAR

- 2K18/SE/041

AIM:- Execute static analysis using cppcheck tool.

THEORY:- Static analysis tools are able to analyze the source code (without running the program) to find problems before they happen. In C/C++ programs, these tools can find program errors like null pointer dereferences, memory leaks, division by zero, integer overflow, out of bounds access, use before initialization, etc.

Cppcheck is a static analysis tool for C/C++ code. It provides unique code analysis to detect bugs and focuses on detecting undefined behaviour and dangerous coding constructs. The goal is to have very few false positives. Cppcheck is designed to be able to analyze our C/C++ code even if it has non-standard syntax (common in embedded projects).

SOURCE CODE For Triangle classification problem (with error):-

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    double a, b, c;
```

```
    double a1, a2, a3;
```

```
    int valid = 0;
```

```
    clrscr();
```

```

printf("Enter first side of the triangle:");

// Enter the sides of Triangle

scanf("%lf", &a);

printf("Enter second side of the triangle:");

scanf("%lf", &b);

printf("Enter third side of the triangle:");

scanf("%lf", &c);

// Checks whether a triangle is valid or not

if (a > 0 && a <= 100 && b > 0 && b <= 100 && c > 0 && c <= 100){

    if ((a + b) > c && (b + c) > a && (c + a) > b)

    {

        valid = 1;

    }

    else

    {

        valid = -1;

    }

}

if (valid == 1)

{

    a1 = (a * a + b * b) / (c * c);

    a2 = (b * b + c * c) / (a * a);

    a3 = (c * c + a * a) / (b * b);

    if (a1 < 1 || a2 < 1 || a3 < 1)

    {

```

```
        printf("Obtuse angled triangle");
    }

    else if (a1 == 1 || a2 == 1 || a3 == 1)
    {
        printf("Right angled triangle");
    }
    else
    {
        printf("Acute angled triangle");
    }
}

else if (valid = -1)    // Here assignment operator is used instead of equality operator
{
    printf("\nInvalid Triangle");
}
else
{
    printf("\nInput Values are Out of Range");
}

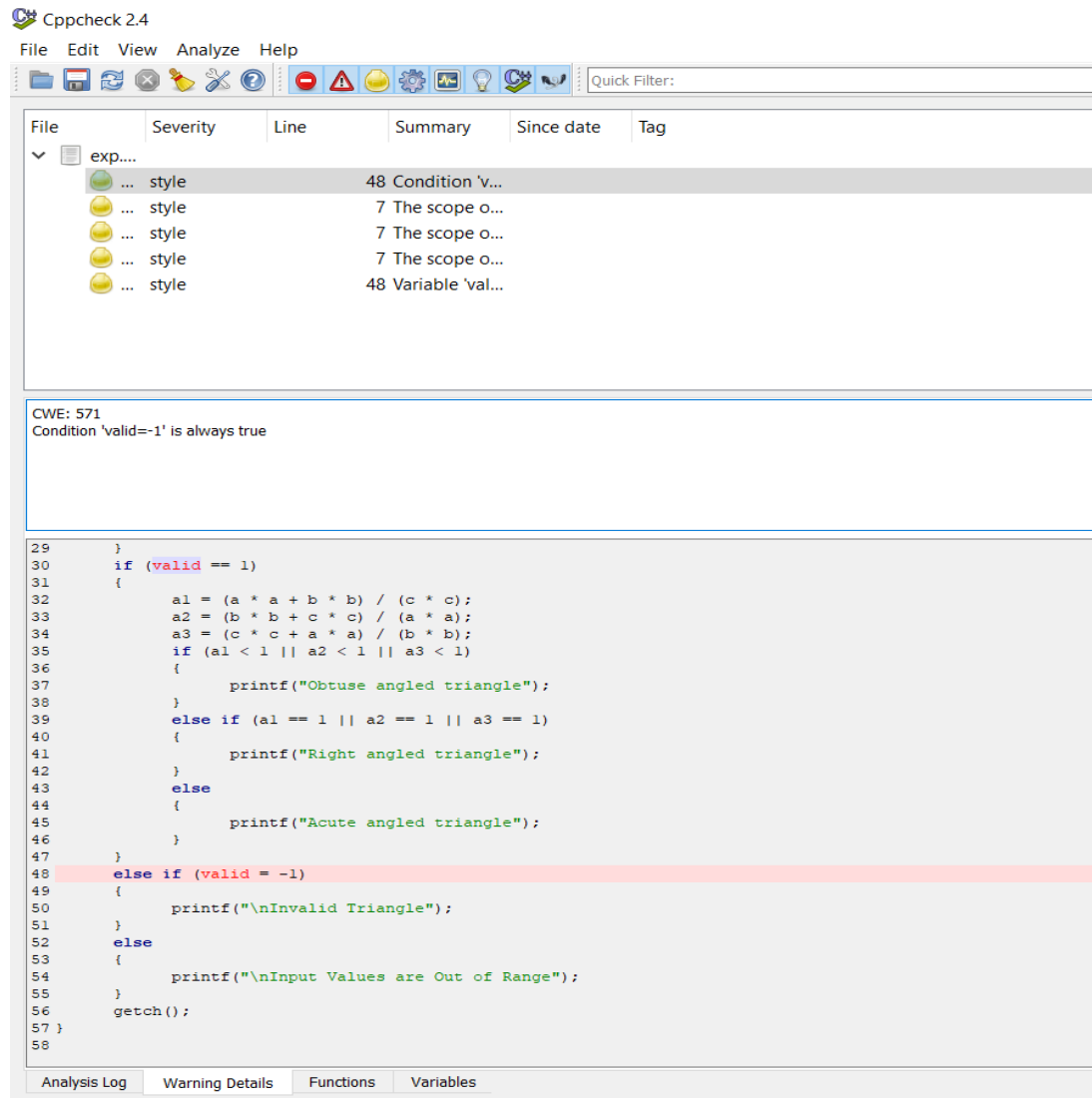
getch();
}
```

OUTPUT:-

Cppcheck shows 5 style warnings which are listed as follows:

1. Condition 'valid= -1' is always true

This warning is detected in line no.48: else if (valid = -1)



The screenshot displays the Cppcheck 2.4 application window. The top menu bar includes File, Edit, View, Analyze, and Help. Below the menu is a toolbar with various icons. A 'Quick Filter:' text box is located on the right side of the toolbar.

The main window is divided into two panes. The top pane shows a list of warnings:

File	Severity	Line	Summary	Since date	Tag
exp...	style	48	Condition 'v...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	48	Variable 'val...		

The bottom pane displays the source code for the file 'exp...'. The code is as follows:

```
29     }
30     if (valid == 1)
31     {
32         a1 = (a * a + b * b) / (c * c);
33         a2 = (b * b + c * c) / (a * a);
34         a3 = (c * c + a * a) / (b * b);
35         if (a1 < 1 || a2 < 1 || a3 < 1)
36         {
37             printf("Obtuse angled triangle");
38         }
39         else if (a1 == 1 || a2 == 1 || a3 == 1)
40         {
41             printf("Right angled triangle");
42         }
43         else
44         {
45             printf("Acute angled triangle");
46         }
47     }
48     else if (valid = -1)
49     {
50         printf("\nInvalid Triangle");
51     }
52     else
53     {
54         printf("\nInput Values are Out of Range");
55     }
56     getch();
57 }
58
```

The warning 'Condition 'valid=-1' is always true' is highlighted in red on line 48. Below the code, there are tabs for 'Analysis Log', 'Warning Details', 'Functions', and 'Variables'.

2. Variable 'valid' is assigned a value that is never used.
This warning is detected in line no. 48: else if (valid = -1)

Cppcheck 2.4

File Edit View Analyze Help

Quick Filter:

File	Severity	Line	Summary	Since date	Tag
exp...	...	48	Condition 'v...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	48	Variable 'val...		

CWE: 563
Variable 'valid' is assigned a value that is never used.

```
29     }
30     if (valid == 1)
31     {
32         a1 = (a * a + b * b) / (c * c);
33         a2 = (b * b + c * c) / (a * a);
34         a3 = (c * c + a * a) / (b * b);
35         if (a1 < 1 || a2 < 1 || a3 < 1)
36         {
37             printf("Obtuse angled triangle");
38         }
39         else if (a1 == 1 || a2 == 1 || a3 == 1)
40         {
41             printf("Right angled triangle");
42         }
43         else
44         {
45             printf("Acute angled triangle");
46         }
47     }
48     else if (valid = -1)
49     {
50         printf("\nInvalid Triangle");
51     }
52     else
53     {
54         printf("\nInput Values are Out of Range");
55     }
56     getch();
57 }
58
```

Analysis Log Warning Details Functions Variables

3. The scope of the variable 'a1' can be reduced.

This warning is detected in line no. 7: double a1, a2, a3;

Cppcheck 2.4

File Edit View Analyze Help

Quick Filter:

File	Severity	Line	Summary	Since date	Tag
exp....	...	48	Condition 'v...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	48	Variable 'val...		

CWE: 398

The scope of the variable 'a1' can be reduced. Warning: Be careful when fixing this message, especially when there are inner loops. Here is an example where cppcheck will write that the scope for 'i' can be reduced:

```
void f(int x)
{
    int i = 0;
    if (x) {
        // it's safe to move 'int i = 0;' here
    }
}
```

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() //Main Begins
5 {
6     double a, b, c;
7     double a1, a2, a3;
8     int valid = 0;
9     clrscr();
10    printf("Enter first side of the triangle:");
11    // Enter the sides of Triangle
12    scanf("%lf", &a);
13    printf("Enter second side of the triangle:");
14    scanf("%lf", &b);
15    printf("Enter third side of the triangle:");
16    scanf("%lf", &c);
17    // Checks whether a triangle is valid or not
18    if (a > 0 && a <= 100 && b > 0 && b <= 100 && c > 0 && c <= 100) {
19
20
21        if ((a + b) > c && (b + c) > a && (c + a) > b)
22        {
23            valid = 1;
24        }
25        else
26        {
27            valid = -1;
28        }
29    }
30    if (valid == 1)
31    {
```

Analysis Log Warning Details Functions Variables

4. The scope of the variable 'a2' can be reduced.

This warning is detected in line no. 7: double a1, a2, a3;

Cppcheck 2.4

File Edit View Analyze Help

Quick Filter:

File	Severity	Line	Summary	Since date	Tag
exp...	...	48	Condition 'v...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	48	Variable 'val...		

CWE: 398
The scope of the variable 'a2' can be reduced. Warning: Be careful when fixing this message, especially when there are inner loops. Here is an example where cppcheck will write that the scope for 'i' can be reduced:
void f(int x)
{
 int i = 0;
 if (x) {
 // it's safe to move 'int i = 0;' here
 }
}

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() //Main Begins
5 {
6     double a, b, c;
7     double a1, a2, a3;
8     int valid = 0;
9     clrscr();
10    printf("Enter first side of the triangle:");
11    // Enter the sides of Triangle
12    scanf("%lf", &a);
13    printf("Enter second side of the triangle:");
14    scanf("%lf", &b);
15    printf("Enter third side of the triangle:");
16    scanf("%lf", &c);
17    // Checks whether a triangle is valid or not
18    if (a > 0 && a <= 100 && b > 0 && b <= 100 && c > 0 && c <= 100) {
19
20
21        if ((a + b) > c && (b + c) > a && (c + a) > b)
22        {
23            valid = 1;
24        }
25        else
26        {
27            valid = -1;
28        }
29    }
30    if (valid == 1)
31    {
```

Analysis Log Warning Details Functions Variables

5. The scope of the variable 'a3' can be reduced.

This warning is detected in line no. 7: double a1, a2, a3;

Cppcheck 2.4

File Edit View Analyze Help

Quick Filter:

File	Severity	Line	Summary	Since date	Tag
exp...	...	48	Condition 'v...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	7	The scope o...		
...	style	48	Variable 'val...		

CWE: 398
The scope of the variable 'a3' can be reduced. Warning: Be careful when fixing this message, especially when there are inner loops. Here is an example where cppcheck will write that the scope for 'i' can be reduced:
void f(int x)
{
 int i = 0;
 if (x) {
 // it's safe to move 'int i = 0;' here
 }
}

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() //Main Begins
5 {
6     double a, b, c;
7     double a1, a2, a3;
8     int valid = 0;
9     clrscr();
10    printf("Enter first side of the triangle:");
11    // Enter the sides of Triangle
12    scanf("%lf", &a);
13    printf("Enter second side of the triangle:");
14    scanf("%lf", &b);
15    printf("Enter third side of the triangle:");
16    scanf("%lf", &c);
17    // Checks whether a triangle is valid or not
18    if (a > 0 && a <= 100 && b > 0 && b <= 100 && c > 0 && c <= 100) {
19
20
21        if ((a + b) > c && (b + c) > a && (c + a) > b)
22        {
23            valid = 1;
24        }
25        else
26        {
27            valid = -1;
28        }
29    }
30    if (valid == 1)
31    {
```

Analysis Log Warning Details Functions Variables