# Workshop 7 (Week 8) – Symbolic Execution

The purpose of this workshop is to practice and develop an understanding of symbolic execution.

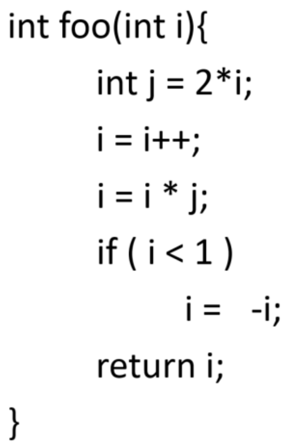
## 1. Concepts

1) What is symbolic execution? What are the differences between concrete execution and symbolic execution?
2) What are the advantages and limitations of symbolic execution?

## 2. The quiz #1

Perform symbolic execution of the following code:

1) What are the path conditions?
2) Generate test cases for each path

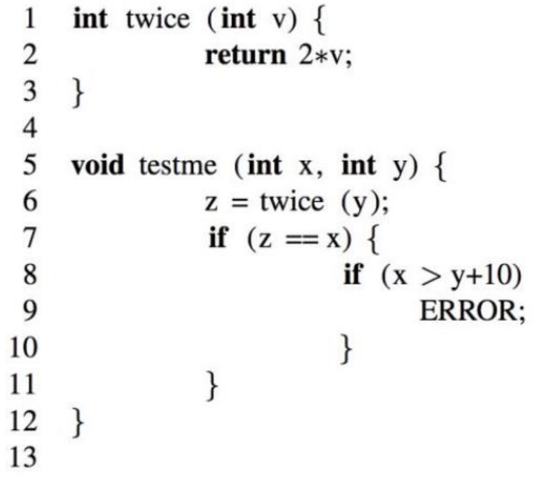


```
int foo(int i){
        int j = 2*i;
        i = i++;
        i = i * j;
        if ( i < 1 )
                i =   -i;
        return i;
}
```

## 3. The quiz #2

Perform symbolic execution of the *testme* function:

1) What are the path conditions?
2) Generate test cases for each path



```
int twice (int v) {
            return 2*v;
}

void testme (int x, int y) {
            z = twice (y);
            if (z == x) {
                        if (x > y+10)
                                    ERROR;
                        }
            }
}
```

## 4. The KLEE Tool

As described in the lecture, KLEE is a symbolic execution tool for C programs. In this workshop you will download and try out KLEE:

- Download and install the KLEE tool from http://klee.github.io/
- Apply KLEE to generate test data for the code shown in Question 3 (Quiz #1) and Question 3 (Quiz #2).
- Apply KLEE to generate test data for the Median program below.

```
// The Median program
int median(int x, int y, int z){
      int median = 0;
      if(x >= y && x <= z){ // y<=x<=z
         median = x;
      } else if(x >= z && x <= y){ // z<=x<=y
         median = x;
      } else if(y >=x && y < z){ // x<=y<=z
         z = y;            // a bug here
      } else if(y >= z && y <= x){ // z<=y<=x
         median = y;
      } else {   // x<=z<=y or y<=z<=x
         median = z;
      }
      return median;
   }
```

## 5. Try the Web: Code In Game

https://www.codingame.com/ide/puzzle/aneo