Lab 6

201301189

1. Created the Control Flow Graph manually and through 2 tools, namely: Control Flow Graph Factory Tool and Eclipse Flow graph generator.

Tool	Whether the CFG drawn by us matches by respective tools
Control Flow graph factory tool	No
Eclipse flow graph generator	Yes

2. Test suites for various coverage criteria:

Coverage criterion	Test vectors: { [x,y]* }+	Expected results
Statement	{ [0,1], [1,0], [2,0] }	min = 2
Branch	{ [1,1], [-1,-1] } , { [-1,-1], [2,-1] }	min = 1
Basic condition	{ [6,5], [4,4], [5,4] }	min = 2

3. Mutation Code:

The mutant that is not killed by any of the test cases given above is the following:

```
for (i = 0; i < p.size(); ++i) {
  if( ((Point) p.get(i).y <))
}
```

The second change can be changing i=0 to i=1 on line 15:

```
for (i = 1; i < p.size(); ++i) {
  if( ((Point) p.get(i).y === null))
}
```

However, there is a problem with this choice. The first loop finds the first point in the vector with the minimum y value, and the second loop is there in order to search through all points with this minimum y and find the one which has the maximum x value. So the first comparison made in the loop (with i=0) is actually always unnecessary — in fact the second loop could start with i=min+1 since min will always be the first point with the minimum y, and the second loop should only be looking at subsequent points min + 1, min + 2 ...

4. Path coverage.

- Both loops do not execute at all.
  `Test cases: {()}`
- The first loop does not execute at all and the second loop executes once.
  `Test cases: {(1,1)}`
- The first loop executes once and the second loop executes twice.
  `Test cases: {(1,2); (2,1)} {(2,1); (1,2)}`

- The first loop executes twice and the second loop executes thrice. Therefore, the total number of test cases required are 28.