CS314 Software Engineering
White Box (Coverage) Testing

Dave Matthews

Colorado State University

Program testing types

- Black-box testing
  - Specifications drive test inputs and expected outputs
  - No code, design or internal documentation are available
- White-box testing
  - Code structure drives test inputs
  - Specification used to derive tests outputs
  - Code, design, and internal documentation are available
  - Goal is to ”cover” the code to gain confidence and detect defects.

White Box Testing

- Statement Coverage
  - Requires all statements to be executed
- Branch Coverage
  - Require decisions evaluate to true and false at least once
  - Implies statement coverage
- Path Coverage
  - Require all possible paths to be executed
  - Implies branch coverage

Coverage Problems

- Statement
  - May not exercise all the conditions in condition predicates
- Branch
  - May not exercise all combinations of branches
- Path
  - combinatorial explosion
  - infinite paths for loops
  - not all paths are feasible

Coverage

- Statements
- Branch
- Path

```
 1 double pow (double x, int y) {
 2   int p = 0;
 3   double z = 1.0;
 4   if (y < 0)
 5     p = -y;
 6   else if (y > 0)
 7     p = y;
 8
 9   while (p-- > 0)
10     z = z * x;
11   if (y < 0)
12     z = 1.0/z;
13   return z;
14 }
```


Statement Coverage

```
 1 double pow (double x, int y) {
 2   int p = 0;
 3   double z = 1.0;
 4   if (y < 0)
 5     p = -y;
 6   else if (y > 0)
 7     p = y;
 8
 9   while (p-- > 0)
10     z = z * x;
11   if (y < 0)
12     z = 1.0/z;
13   return z;
14 }
```

```
// 1,2,3,4,6,7,9,10,11,13
pow(2,2);

// 1,2,3,4,5,9,10,11,12,13
pow(2,-2);
```

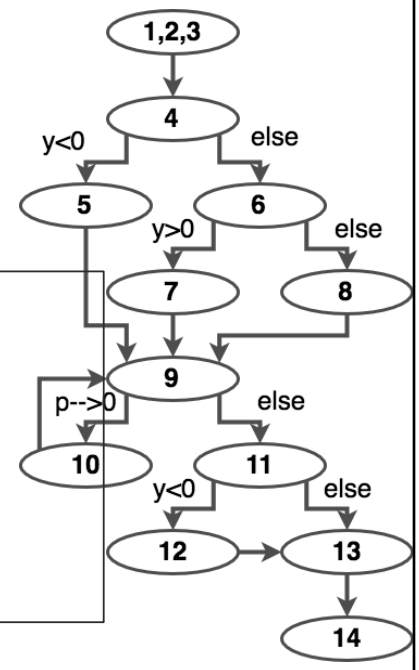
Branch Coverage

```
 1 double pow (double x, int y) {
 2   int p = 0;
 3   double z = 1.0;
 4   if (y < 0)
 5     p = -y;
 6   else if (y > 0)
 7     p = y;
 8   ; else nothing
 9   while (p-- > 0)
10     z = z * x;
11   if (y < 0)
12     z = 1.0/z;
13   return z;
14 }
```

```
// 4F,6T,9T,9F,11F
pow(2,2);

// 4T,9T,9F,11T
pow(2,-2);

// 4F,6F,9F,11F
pow(2,0);
```



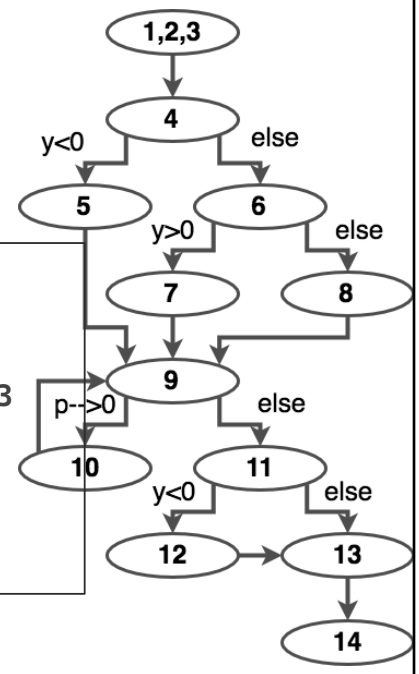
Path Coverage

```
 1 double pow (double x, int y) {
 2   int p = 0;
 3   double z = 1.0;
 4   if (y < 0)
 5     p = -y;
 6   else if (y > 0)
 7     p = y;
 8   ; else nothing
 9   while (p-- > 0)
10     z = z * x;
11   if (y < 0)
12     z = 1.0/z;
13   return z;
14 }
```

```
// 1,2,3,4,6,7,9,10,11,13
pow(2,2);

// 1,2,3,4,5,9,10,11,12,13
pow(2,-2);

// 1,2,3,4,6,8,9,11,13
pow(2,0);
```



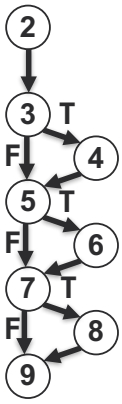
Statement Coverage

```
 1 public class Example {
 2   public int xmpl( int x, boolean c1, boolean c2, boolean c3) {
 3     if (c1)
 4       x++;
 5     if (c2)
 6       x--;
 7     if (c3)
 8       x = -x;
 9     return x;
10   }
11 }
```

```
// 2,3,4,5,6,7,8,9
xmpl(9, true, true, true);
```


Branch Coverage

```
 1 public class Example {
 2   public int xmpl( int x, boolean c1, boolean c2, boolean c3) {
 3     if (c1)
 4       x++;
 5     if (c2)
 6       x--;
 7     if (c3)
 8       x = -x;
 9     return x;
10   }
11 }
```

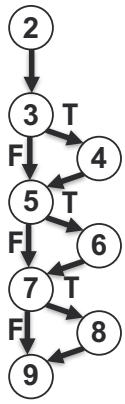



```
// 3T,5T,7T
xmpl(9, true, true, true);

// 3F,5F,7F
xmpl(7, false, false, false);
```

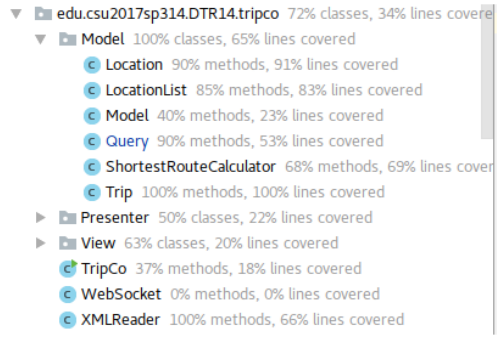
Path Coverage

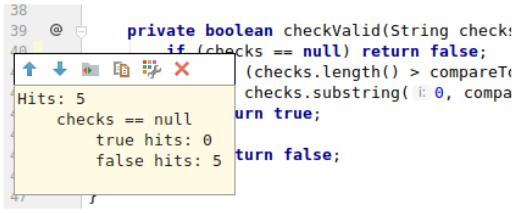
```
 1 public class Example {
 2   public int xmpl( int x, boolean c1, boolean c2, boolean c3) {
 3     if (c1)
 4       x++;
 5     if (c2)
 6       x--;
 7     if (c3)
 8       x = -x;
 9     return x;
10   }
11 }
```




```
// 2,3,4,5,6,7,8,9
xmpl(9, true, true, true);
// 2,3,5,7,9
xmpl(9, false, false, false);
// 2,3,4,5,6,7,9
xmpl(9, true, true, false);
// 2,3,4,5,7,8,9
xmpl(9, true, false, true);
// 2,3,4,5,7,9
xmpl(9, true, false, false);
// 2,3,4,6,7,8,9
xmpl(9, false, true, true);
// 2,3,5,6,7,9
xmpl(9, false, true, false);
// 2,3,5,7,8,9
xmpl(9, false, false, true);
```


IntelliJ IDEA Coverage





[all classes]

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	72.7% (16/ 22)	41.1% (81/ 197)	34.8% (560/ 1609)

Coverage Breakdown

Package	Class, %	Method, %	Line, %
edu.csu2017sp314.DTR14.tripco	66.7% (2/ 3)	19.4% (6/ 31)	11% (27/ 245)
edu.csu2017sp314.DTR14.tripco.Model	100% (6/ 6)	76.6% (49/ 64)	65% (368/ 566)
edu.csu2017sp314.DTR14.tripco.Presenter	50% (1/ 2)	37.5% (3/ 8)	22.2% (10/ 45)
edu.csu2017sp314.DTR14.tripco.View	63.6% (7/ 11)	24.5% (23/ 94)	20.6% (155/ 753)

IntelliJ IDEA Coverage

- https://www.jetbrains.com/help/idea/code-coverage.html
- https://www.youtube.com/watch?v=yNYzZvyA2ik

Code Climate Coverage

Issues Source Stats

Issues Coverage

```
1  package com.tripco.t00.planner;
2
3  import com.google.gson.Gson;
4  import com.google.gson.JsonElement;
5  import com.google.gson.JsonParser;
6  import com.tripco.t00.server.HTTP;
7  import spark.Request;
8  import java.util.ArrayList;
9
10 /**
11  * The Trip class supports TFFI so it can easily be converted to/from Json by Gson.
12  *
13  */
14 public class Trip {
15   // The variables in this class should reflect TFFI.
16   public String type;
17   public String title;
18   public Option options;
19   public ArrayList<Place> places;
20   public ArrayList<Integer> distances;
21   public String map;
22
23   /** The top level method that does planning.
24    * At this point it just adds the map and distances for the places in order.
25    * It might need to reorder the places in the future.
26    */
27   public void plan() {
28
29     this.map = svg();
30     this.distances = legDistances();
31
32   }
33
```

```
18 public class Plan {
19
20   private Trip trip;
21
22   /** Handles trip planning request, creating a new trip object from the trip request.
23    * Does the conversion from Json to a Java class before planning the trip.
24    * @param request
25    */
26   public Plan (Request request) {
27     // first print the request
28     System.out.println(HTTP.echoRequest(request));
29
30     // extract the information from the body of the request.
31     JsonParser jsonParser = new JsonParser();
32     JsonElement requestBody = jsonParser.parse(request.body());
33
34     // convert the body of the request to a Java class.
35     Gson gson = new Gson();
36     trip = gson.fromJson(requestBody, Trip.class);
37
38     // plan the trip.
39     trip.plan();
40
41     // log something.
42     System.out.println(trip.title);
43   }
44
45   /** Handles the response for a Trip object.
46    * Does the conversion from a Java class to a Json string.*
47    */
48   public String getTrip () {
49     Gson gson = new Gson();
50     return gson.toJson(trip);
51   }
52 }
```