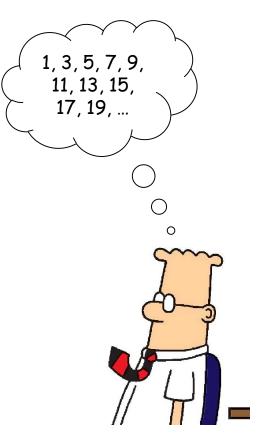
CSc 352: Testing and Code Coverage

Testing and test cases

```
int main()
                                    make sure you
                                     use at least
                                    fifty different
                                      test inputs
  read x;
  if (x is odd) {
     compute payroll data;
  else {
     delete all files;
     send rude email to boss;
     crash computer;
```



Testing and test cases

```
int main()
                                     make sure you
                                                                    1, 3, 5, 7, 9,
                                                                     11, 13, 15,
                                      use at least
                                     fifty different
                                                                      17, 19, ...
                                       test inputs
  read x;
  if (x is odd) {
     compute payroll data;
  else {
     delete all files;
                                                It isn't enough to have a lot
     send rude email to boss;
                                                 of test cases. We have to
                                                make sure our tests "cover"
     crash computer;
                                                 the program adequately.
```

gcov: a code coverage analyzer

- test coverage program
 - indicates how many times each line was executed
 - marks code that did not get executed
 - cumulative over a set of tests
- helps you understand
 - how effectively your current test cases "cover" the code
 - what additional test inputs you need in order to get better coverage
 - needs the program to be compiled with additional gcc options

```
% gcc -Wall -fprofile-arcs -ftest-coverage fib.c
#include <stdio.h>
                                                 % ./a.out
                                       input
int fib(int n)
                                                 fib(2) is 1
                                                 % ./a.out
 if (n <= 0) {
   return 0;
                                        input
                                                 fib(3) is 2
 else if (n == 1) {
                                                 % ./a.out
   return 1;
                                        input
                                                 fib(4) is 3
 else {
   int i, f, f0 = 0, f1 = 1;
                                                 % ./a.out
                                        input
   for (i = 1; i < n; i++) {</pre>
                                                 fib(5) is 5
     f = f0+f1;
                                                 % ./a.out
     f0 = f1:
                                       input
     f1 = f;
                                                 fib(6) is 8
                                                 % gcov fib.c
   return f;
                                                 File 'fib.c'
                                                 Lines executed:86.67% of 15
int main()
                                                        testing didn't
 int n;
                                                         execute all of the
 scanf("%d", &n);
                                                        code
 printf("fib(%d) is %d\n", n, fib(n));
 return 0;
```

additional compiler flags

```
% more fib.c.gcov
                                                                           0:Source:fib.c
                                                                     -:
                                                        % gc
                                                                           0:Graph:fib.gcno
#include <stdio.h>
                                                        % ./
                                                                           0:Data:fib.gcda
                                                                           0:Runs:1
                                             input
int fib(int n)
                                                                           0:Programs:1
                                                        fib(
                                                                           1:
                                                        % ./
  if (n <= 0) {
                                                                           2:
                                                                           3:#include <stdio.h>
                                                        3
                                                                     -:
    return 0;
                                             input
                                                                     -:
                                                        fib(
                                                                           5: int fib(int n)
                                                                     1:
  else if (n == 1) {
                                                        % ./
                                                                           6:
    return 1;
                                              input
                                                        4
                                                                                 if (n <= 0) {
                                                                     1:
                                                                           7:
                                                        fib(
                                                                  ####:
                                                                                   return 0;
  else {
                                                        % ./
    int i, f, f0 = 0, f1 = 1;
                                                                                 else if (n == 1
                                                                          10:
                                                                     1:
                                                        5
                                              input
                                                                 #####:
                                                                          11:
                                                                                   return 1;
    for (i = 1; i < n; i++) {
                                                        fib(
                                                                          12.
      f = f0+f1;
                                                        % ./
                                                                          13:
                                                                                 else {
                                                                     -:
      f0 = f1:
                                                                                   int i, f, f0 = 0, f1 = 1;
                                                                     1:
                                                                          14:
                                             input
                                                        6
      f1 = f;
                                                                     -:
                                                                          15:
                                                        fib(
                                                                     8:
                                                                          16:
                                                                                   for (i = 1; i < n; i++) {
                                                        % gc
                                                                          17:
                                                                                     f = f0+f1;
                                                                     7:
    return f;
                                                                                     f0 = f1;
                                                                          18:
                                                        File
                                                                     7:
                                                                     7:
                                                                          19:
                                                                                     f1 = f;
                                                        Line
                                                                     -:
                                                                          20:
                                                                          21:
                                                                     -:
int main()
                                                                     1:
                                                                          22:
                                                                                   return f;
                                                                          23:
  int n;
                                                                     -:
                                                                          24:
                                                                     -:
                                                                          25:
  scanf("%d", &n);
                                                                               int main()
                                                                     1:
                                                                          26:
                                                                     -:
                                                                          27:
  printf("fib(%d) is %d\n", n, fib(n));
                                                                     -:
                                                                          28:
                                                                                 int n;
                                                                          29:
                                                                     -:
                                                                                 scanf("%d", &n);
                                                                     1:
                                                                          30:
  return 0;
                                                                     -:
                                                                          31:
                                                                                 printf("fib(%d) is %d\n", n, fib(n));
                                                                     1:
                                                                          32:
                                                                     -:
                                                                          33:
                                                                     1:
                                                                           34:
                                                                                 return 0;
                                                                          35: }
                                                                     -:
```

6

```
#include <stdio.h>
int fib(int n)
 if (n <= 0) {
    return 0;
  else if (n == 1) {
    return 1;
  else {
   int i, f, f0 = 0, f1 = 1;
    for (i = 1; i < n; i++) {
     f = f0+f1;
     f0 = f1;
      f1 = f;
    return f;
int main()
 int n;
  scanf("%d", &n);
  printf("fib(%d) is %d\n", n, fib(n));
  return 0;
```

input

```
% ./a.out
1
fib(1) is 1
% gcov fib.c
fib.c:source file is newer than notes file 'fib.gcno'
(the message is only displayed one per source file)
File 'fib.c'
Lines executed:93.33% of 15
Creating 'fib.c.gcov'
```

```
% more fib.c.gcov
                                                                             0:Source:fib.c
                                                                             0:Graph:fib.gcno
                                                                             0:Data:fib.gcda
                                                      % ./a.o
                                                                             0:Runs:2
#include <stdio.h>
                                                                             0:Programs:1
                                           input
                                                                             0:Source is newer than graph
int fib(int n)
                                                      fib(1)
                                                                       -:
                                                                             1:
                                                      % gcov
                                                                       -:
                                                                             2:
  if (n <= 0) {
                                                                             3: #include <stdio.h>
                                                                       -:
                                                      fib.c:s
    return 0;
                                                                       -:
                                                                             4:
                                                      (the me
                                                                             5: int fib(int n)
                                                                       2:
  else if (n == 1) {
                                                      File 'f
                                                                             6: {
    return 1:
                                                      Lines e
                                                                       2:
                                                                             7:
                                                                                   if (n <= 0) {
                                                                                     return 0;
                                                                    ####:
                                                      Creatin
  else {
    int i, f, f0 = 0, f1 = 1;
                                                                            10:
                                                                                   else if (n == 1) {
                                                                       2:
                                                                            11:
                                                                       1:
                                                                                     return 1;
    for (i = 1; i < n; i++) {
                                                                            12:
                                                                       -:
                                                                            13:
      f = f0+f1;
                                                                       -:
                                                                                   else {
                                                                                    int i, f, f0 = 0, f1 = 1;
      f0 = f1:
                                                                       1:
                                                                            14:
                                                                            15:
                                                                       -:
      f1 = f;
                                                                       8:
                                                                            16:
                                                                                     for (i = 1; i < n; i++) {
                                                                       7:
                                                                            17:
                                                                                       f = f0+f1;
                                                                            18:
                                                                                       f0 = f1;
    return f;
                                                                       7:
                                                                            19:
                                                                                       f1 = f:
                                                                       -:
                                                                            20:
                                                                       -:
                                                                            21:
                                                                       1:
                                                                            22:
                                                                                     return f;
int main()
                                                                       -:
                                                                            23:
                                                                            24: }
  int n;
                                                                       -:
                                                                            25:
                                                                       2:
                                                                            26: int main()
  scanf("%d", &n);
                                                                            27:
                                                                                {
                                                                       -:
                                                                            28:
                                                                                   int n;
                                                                       -:
  printf("fib(%d) is %d\n", n, fib(n));
                                                                            29:
                                                                       -:
                                                                                   scanf("%d", &n);
                                                                       2:
                                                                            30:
                                                                       -:
                                                                            31:
  return 0;
                                                                       2:
                                                                            32:
                                                                                   printf("fib(%d) is %d\n", n, fib(n));
                                                                       -:
                                                                            33:
                                                                       2:
                                                                            34:
                                                                                   return 0;
                                                                            35: }
```

```
#include <stdio.h>
int fib(int n)
 if (n <= 0) {
    return 0;
  else if (n == 1) {
    return 1;
  else {
    int i, f, f0 = 0, f1 = 1;
    for (i = 1; i < n; i++) {
      f = f0+f1;
      f0 = f1:
      f1 = f;
    return f;
int main()
 int n;
  scanf("%d", &n);
  printf("fib(%d) is %d\n", n, fib(n));
  return 0;
```

input

```
% ./a.out
0
fib(0) is 0
% gcov fib.c
fib.c:source file is newer than notes file 'fib.gcno'
(the message is only displayed one per source file)
File 'fib.c'
Lines executed:100.00% of 15
creating 'fib.c.gcov'
```

Code coverage and testing

- Just because every line has been executed does not mean the program has been tested thoroughly
 - we may want to test the same line of code under different conditions
 - e.g.: a loop should be tested with values that cause 0, 1, and "many" iterations
- However, if some lines are not executed the program is definitely not thoroughly tested
 - gcov helps us identify and fix this
 - exception: "system errors" that may be difficult to create

Example of not enough testing

```
eanson@lectura:~/oldHome/inClass$ cat upper.c
#include <stdio.h>
void myup(char c[]) {
  int i:
  for (i=0; c[i] != '\0'; ++i) {
    if (c[i] > 'a' \&\& c[i] < 'z')
    c[i] = c[i] - 'a' + 'A';
  return;
int main () {
  char st[64];
  scanf ("%63s", st);
  myup(st);
  printf("%s\n", st);
  return 0;
eanson@lectura:~/oldHome/inClass$
```

This is (almost) the program we wrote in class to convert all lower case letters in a string to upper case.

Example of not enough testing

```
eanson@lectura:~/oldHome/inClass$ cat upper.c
#include <stdio.h>
void myup(char c[]) {
  int i;
  for (i=0; c[i] != '\0'; ++i) {
    if (c[i] > 'a' \&\& c[i] < 'z')
    c[i] = c[i] - 'a' + 'A';
  return;
                                            Compile and test it. 100% of
int main () {
                                            code is executed and the
  char st[64];
                                            result is correct. So the code
  scanf("%63s", st);
                                            has no bugs, right?
  myup (st);
  printf("%s\n", st);
  return 0;
        $gcc -fprofile-arcs -ftest-coverage -Wall -o upper upper.c
eanson@
        $upper
        ThisTest1.
        THISTEST1.
        $qcov upper.c
        File 'upper.c'
        Lines executed: 100.00% of 10
        upper.c:creating 'upper.c.gcov'
```

Example of not enough testing

```
eanson@lectura:~/oldHome/inClass$ cat upper.c
#include <stdio.h>
                                                There were still
void myup(char c[]) {
                                                errors.
  int i;
  for (i=0; c[i] = (0; +\pm i)
    if (c[i]( >)'a' && c[i]( <
  return;
                                            Compile and test it. 100% of
int main () {
                                            code is executed and the
  char st[64];
                                            result is correct. So the code
  scanf ("%63s", st);
                                            has no bugs, right?
 myup (st);
  printf("%s\n", st);
  return 0;
                                                    Oops!
eanson@lectura:~/oldHome/inClass$
                               $upper
                               San Diego zo
                               SaN DIEGO z
```

gcov: summary

- code coverage testing tool
 - works with gcc; needs additional compiler flags
 - gcc –fprofile-arcs –ftest-coverage ...
- shows execution frequency of each line of code
 - reports % of lines covered by tests
 - coverage values are cumulative
 - delete *.gcda file to start afresh
 - how many times each line was executed
 - highlights lines not executed