

JTH, LSH, MDE

Randy sez this is the most important  
"paper" he's written - can we / should we adopt it?

## Tips for Working Successfully in a Group

By Randy Pausch, for the Building Virtual Worlds course at Carnegie Mellon, Spring 1998

**Meet people properly.** It all starts with the introduction. Then, exchange contact information, and make sure you know how to pronounce everyone's names. Exchange phone #s, and find out what hours are acceptable to call during.

**Find things you have in common.** You can almost always find something in common with another person, and starting from that baseline, it's much easier to then address issues where you have differences. This is why cities like professional sports teams, which are socially galvanizing forces that cut across boundaries of race and wealth. If nothing else, you probably have in common things like the weather.

**Make meeting conditions good.** Have a large surface to write on, make sure the room is quiet and warm enough, and that there aren't lots of distractions. Make sure no one is hungry, cold, or tired. Meet over a meal if you can; food *softens* a meeting. That's why they "do lunch" in Hollywood.

**Let everyone talk.** Even if you think what they're saying is stupid. Cutting someone off is rude, and not worth whatever small time gain you might make. Don't finish someone's sentences for him or her; they can do it for themselves. And remember: talking louder or faster doesn't make your idea any better.

**Check your egos at the door.** When you discuss ideas, immediately label them and write them down. The labels should be descriptive of the idea, not the originator: "the troll bridge story," not "Jane's story."

**Praise each other.** Find something nice to say, even if it's a stretch. Even the worst of ideas has a silver lining inside it, if you just look hard enough. Focus on the good, praise it, and then raise any objections or concerns you have about the rest of it.

**Put it in writing.** Always write down who is responsible for what, by when. Be *concrete*. Arrange meetings by email, and establish accountability. Never assume that someone's roommate will deliver a phone message. Also, remember that "politics is when you have more than 2 people" - with that in mind, always CC (carbon copy) any piece of email within the group, or to me, to all members of the group. This rule should *never* be violated; don't try to guess what your group mates might or might not want to hear about.

**Be open and honest.** Talk with your group members if there's a problem, and talk with me if you think you need help. The whole point of this course is that it's tough to work across cultures. If we all go into it knowing that's an issue, we should be comfortable discussing problems when they arise -- after all, that's what this course is really about. Be forgiving when people make mistakes, but don't be afraid to raise the issues when they come up,

**Avoid conflict at all costs.** When stress occurs and tempers flare, take a short break. Clear your heads, apologize, and take another stab at it. Apologize for upsetting your peers, even if you think someone else was primarily at fault; the goal is to work together, not start a legal battle over whose transgressions were worse. It takes two to have an argument, so be the peacemaker.

**Phrase alternatives as questions.** Instead of "I think we should do A, not B," try "What if we did A, instead of B?" That allows people to offer comments, rather than defend one choice.

## CHAPTER 1

### A Self-Assessment Test

Before beginning the book, it is strongly recommended that you take the following short test. The problem is the testing of the following program:

The program reads three integer values from a card. The three values are interpreted as representing the lengths of the sides of a triangle. The program prints a message that states whether the triangle is scalene, isosceles, or equilateral.

On a sheet of paper, write a set of test cases (i.e., specific sets of data) that you feel would adequately test this program. When you have completed this, turn the page to analyze your tests.

staff  
in-  
re-  
ngt-  
He  
at  
ork  
ind  
in  
th  
u-  
re  
n-

## 2 A SELF-ASSESSMENT TEST

The next step is an evaluation of the effectiveness of your testing. It turns out that this program is more difficult to write than it first appears to be. As a result, different versions of this program have been studied, and a list of common errors has been compiled. Evaluate your set of test cases by using it to answer the following questions. Give yourself one point for each "yes" answer.

1. Do you have a test case that represents a *valid* scalene triangle? (Note that test cases such as 1,2,3 and 2,5,10 do not warrant a "yes" answer, because there does not exist a triangle having such sides.)
2. Do you have a test case that represents a valid equilateral triangle?
3. Do you have a test case that represents a valid isosceles triangle? (A test case specifying 2,2,4 would not be counted.)
4. Do you have at least three test cases that represent valid isosceles triangles such that you have tried all three permutations of two equal sides (e.g., 3,3,4; 3,4,3; and 4,3,3)?
5. Do you have a test case in which one side has a zero value?
6. Do you have a test case in which one side has a negative value?
7. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is equal to the third? (That is, if the program said that 1,2,3 represents a scalene triangle, it would contain a bug.)
8. Do you have at least three test cases in category 7 such that you have tried all three permutations where the length of one side is equal to the sum of the lengths of the other two sides (e.g., 1,2,3; 1,3,2; and 3,1,2)?
9. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is less than the third (e.g., 1,2,4 or 12,15,30)?
10. Do you have at least three test cases in category 9 such that you have tried all three permutations (e.g., 1,2,4; 1,4,2; and 4,1,2)?
11. Do you have a test case in which all sides are 0 (i.e., 0,0,0)?
12. Do you have at least one test case specifying noninteger values?
13. Do you have at least one test case specifying the wrong number of values (e.g., two, rather than three, integers)?
14. For each test case, did you specify the expected output from the program in addition to the input values?

Of course, a set of test cases that satisfies the above conditions does not guarantee that all possible errors would be found,

C  
a  
p  
si  
pi  
sc  
ve  
At  
se  
in  
wa  
Th  
er  
the  
po  
Wh  
on  
fec  
siv  
nor  
unc  
adc  
TE  
tion  
ger  
er-c  
insp  
nor  
thro  
from  
poin  
dura  
THE  
cont  
niqu  
ples  
prog  
comj  
the fi  
self-a  
date  
the fi  
"If the  
one n  
of the  
many

but si  
currec  
this p  
have  
exper  
only 7  
trate  
an ea  
a 100  
even :

A SELF-ASSESSMENT TEST 3

but since questions 1-13 represent errors that have actually occurred in different versions of this program, an adequate test of this program should expose these errors. If you are typical, you have done poorly on this test. As a point of reference, highly experienced professional programmers score, on the average, only 7.8 out of a possible 14. The point of the exercise is to illustrate that the testing of even a trivial program such as this is not an easy task. And if this is true, consider the difficulty of testing a 100,000-statement air-traffic-control system, a compiler, or even a mundane payroll program.

staff  
in-  
re-  
engi-  
He  
at  
ork  
and  
in  
ch-  
au-  
are  
m-

effectiveness of your  
more difficult to write  
different versions of this  
common errors has been  
by using it to answer  
a point for each "yes"

sents a *valid* scalene  
s 1,2,3 and 2,5,10 do not  
here does not exist a

nts a valid equilateral

sents a valid isosceles  
would not be counted.)  
s that represent valid  
e tried all three permu-  
,4,3; and 4,3,3)?  
ide has a zero value?  
ne side has a negative

integers greater than  
umbers is equal to the  
that 1,2,3 represents a  
g.)  
in category 7 such that  
ns where the length of  
ngths of the other two

integers greater than  
e numbers is less than

in category 9 such that  
ns (e.g., 1,2,4; 1,4,2; and

ides are 0 (i.e., 0,0,0)?  
specifying noninteger

e specifying the wrong  
an three, integers)?  
e expected output from  
values?

tisfies the above condi-  
errors would be found,