Teaching Black Box Testing to High School Students

Amitrajit Sarkar
Christchurch Polytechnic Institute of Technology
New Zealand

Tim Bell
University of Canterbury
New Zealand
CS Field guide

Find your way to Suburbapolis Station!
You are currently at:

City Mall Station

Do you want to travel on:
Train A or Train B

Start Over

Black box testing
computerscienceclub.org
AS91636

- Demonstrate understanding of areas of computer science

- Level 3 (Year 13)
AS91636 (level 3)

- describing **key problems** that are addressed in selected **areas** of computer science
- discussing examples of **practical applications** of selected areas to demonstrate the use of key algorithms and/or techniques
- evaluating the **effectiveness** of algorithms, techniques, or applications
AS91636 (level 3)

- formal languages
- network communication protocols
- complexity and tractability
- intelligent systems
- software engineering
- graphics and visual computing
Achievement Standard

Subject Reference: Digital Technologies 3.44
Title: Demonstrate understanding of areas of computer science
Level: 3
Credits: 4
Assessment: External
Subfield: Technology
Domain: Digital Technologies
Status: Registered
Status date: 4 December 2012
Planned review date: 31 December 2016
Date version published: 4 December 2012
Goals

- Grounding in CS concepts
- Discover a passion for computing
Software testing

- Not debugging
- Black box testing
  - Functional testing
  - User acceptance testing
Importance of testing

- Programming 20% of projects
- About 25% (1999) to 33% (2009) of projects succeed
- ¼ failed outright or cancelled (2009)
- Testing 50% of cost
- Top-of-mind issue for managers
- MS: One tester per programmer
- Growing in ACM curriculum
Testing teaching

- Florida IT testing course
- Georgia tech
- NSF funded courses
- testing-challenges.org
int myValue=112;
int* myPtr=&myValue;
myValue=77;
cout<<"My new value is ";
cout<<myValue<<endl;
cout<<"I can access my new value ";
<<"using a pointer"<<endl;
cout<<"The value is ";
cout<<myPtr;
16. Software engineering

Warning: This chapter is still in development; the current version has been released for the 2013 school year with enough material to enable students to complete the NZ 3.44 achievement standard, but more material may be added or updated during the year to give more options for student projects.

16.1. What's the big picture?

Software failures happen all the time. Sometimes it's a little bug that makes a program difficult to use; other times an error might crash your entire computer. Some software failures are more spectacular than others.

In 1996, The ARIANE 5 rocket of the European Space Agency was launched for its first test flight: Countdown, ignition, flame and smoke, soaring rocket... then BANG! Lots of little pieces scattered through the South American rainforest.
Initial testing experience

Amount is incorrect. Please enter only numbers. Decimals are allowed up to two places.

Amount

$702.40

Amount is incorrect. Please enter only numbers. Decimals are allowed up to two places.

Amount

662.40
Black box testing

Initial testing experience
Meyer's triangle problem

http://practice.agilistry.com/triangle
System with errors

This test will determine what triangle you have made depending on the integer values in the input fields.

Enter the side values of a triangle below:

**Side 1:**

2

**Side 2:**

3

**Side 3:**

3

Submit

This is Scalene triangle.
We want you to write a set of test cases—specific sets of data—to properly test a relatively simple program. Create a set of test data for the program—data the program must handle correctly to be considered a successful program. Here's a description of the program:

1. The Triangle program accepts three integer values as input. Each value represents a side of the triangle.
2. If the inputs are invalid (sides smaller than 0, or not integers) or if fewer than three values are provided the program outputs the message "Invalid input value(s)".
3. If the length of the largest side is greater or equal to the sum of the lengths of the two smaller sides the program will output the message "Not a Triangle".
4. If all three sides of the triangle are of equal length the program will output the message "Equilateral".
5. If exactly two sides of the triangle are of equal length the program will output the message "Isosceles".
6. If all three sides of the triangle are of different lengths the program will output the message "Scalene".

You have met 6 out of 10 criteria.

1. must have one test case with all sides equal (and non zero)
2. must have one test case with all sides equal (and zero)
3. must have one test case with no sides equal (and no zero sides)
4. no side must be greater than the sum of the other two
5. no sides should be negative
6. Must have one right angled triangle

| 1, 1, 1, equi |
| 3/4, 6, go |
| 0, 0, 0, e |
| 3, 3, 4, s |
Development

- Iterative improvement
- Short explanations
- Warm-up examples
- Use program first
- Teacher guidance
Ideas to test an alarm clock in 30 minutes

Assuming that this is the only question put forth of us and we cannot get any more information on the mission like time required to run tests, how much money we could spend on testing, who the product end users are, is it a handheld alarm clock or a digital one, etc. This would be our generic approach considering that the alarm clock is a physical device with only 2 buttons one for the snooze and one for the stop as displayed in the picture below.
Personalised report

- **Key problem**: assure product (digital alarm clock) suitable for the end user.
- **Key algorithms or techniques**: black-box testing (and acceptance testing)
- **Practical application**: software engineering projects
- **Evaluation**: critically evaluate an existing product (the alarm clock).
Lesson:

- How to test
- Testing is hard
- Testing is important in SE
csfieldguide.org.nz

tim.bell@canterbury.ac.nz