

CS 458 - Homework 10

Deadline

Due by 11:59 pm on Friday, December 2, 2016

How to submit

Submit your file using `~st10/458submit` on nrs-projects, with a homework number of 10.

Purpose

To consider the distinction between verification and validation, and to consider some testing concepts from Jalote Chapter 8.

Important notes

- Note that some of your submissions for this assignment may be posted to the course Moodle site.
- Create a file named `458hw10.txt` that starts with your name. Then give the problem number, the problem part, and your answer(s) for each of the following problems.

Problem 1

Consider the following definitions from Wikipedia, citing IEEE-STD-610:

Verification - "The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase."

Validation - "The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements."

These are not quite the same thing, and I think it is important to understand the distinction -- one way I have heard this described is:

- verification ensures 'you built it right', it meets the specifications;
- validation ensures 'you built the right thing', it actually meets the user's needs, meets the requirements (and that the specifications were correct in the first place)

1 part a

Is unit testing of a module more accurately characterized as verification or validation?

1 part b

Is acceptance testing of a module more accurately characterized as verification or validation?

1 part c

Is a requirements review (as described in Jalote Chapter 3, p. 65) more accurately characterized as verification or validation?

1 part d

Is a code inspection (as describe in Jalote Chapter 7, section 7.5) more accurately characterized as verification or validation?

Problem 2

Consider the following fragment of code from Homework 9, and then answer the questions that follow.

```
function lin_search(A, E): boolean
var
    i: integer;
    found: boolean;
begin
    found := false;
    i := 0;
    while (not found) and (i < A.length) do begin
        if (A[i] = E) then
            found := true;
        i := i + 1;
    end;
    lin_search := found;
end;
```

2 part a

Assume that you test `lin_search` JUST with an empty array as its first argument array and the value 12 as its second argument.

Would you have complete **statement** coverage?

Would you have complete **branch** coverage?

2 part b

Now assume you test `lin_search` JUST with an array containing 3, 6, and 9 as its first argument array and the value 3 as its second argument.

Would you have complete **statement** coverage?

Would you have complete **branch** coverage?

2 part c

Now assume you test `lin_search` by running it four times,

- once with an empty array as its first argument array and the value 12 as its second argument,
- once with an array containing 3, 6, and 9 as its first argument array and the value 3 as its second argument,
- once with an array containing 2, 4, 6, 8 as its first argument array and the value 6 as its second argument,
- and once with an array containing 1, 3, 5, 7, 9 as its first argument array and the value 20 as its second argument.

Would you have complete **statement** coverage?

Would you have complete **branch** coverage?

Problem 3**3 part a**

If you design your test cases trying to obtain complete statement coverage, is that considered to be black box testing or white box testing?

3 part b

If you design your test cases trying to obtain complete branch coverage, is that considered to be black box testing or white box testing?

3 part c

If you design your test cases by trying to determine what the different categories of possible input data are, and making sure that you include test cases including each category of input data, is that considered to be black box testing or white box testing?

Submit your resulting file `458hw10-1.txt`.