**CS 4321 - Video Lecture Expectations**

**Video P4L2: Black Box Testing (46 minutes)**

Watch video P4L2 on Udacity. Watch clips 1-20; you may omit clips 21-25. Answer the questions below and submit on Blazeview (HW-P4L2). Instructions:

* Do not remove the questions.
* You can provide the answer(s) where the blank is, but preserve the underline (or use a different color for the answers)
* Or, you can provide the answers below the questions. For example, you could type: Answer: x, y, z.

Clips 14-20 are especially important technique (category partition method). Frequently, I generally give two homework assignments on this technique. I’m not sure about this semester. If so, it will be later in the semester.

**Questions to be answered**

1. Black-box testing is also called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. List four advantages of black-box testing.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. A systematic approach to black-box testing starts with a specification of the software which is used to identify \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ which are used to identify \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ from which we derive \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and finally we generate specific \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. The problem of identifying relevant inputs for some software or some feature of it is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ testing is not feasible even for the most trivial pieces of code.
4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ refers to making assumptions about the behavior of a user of a piece of code will be used, for example, assuming the user will not use the software incorrectly.
5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ testing eliminates designer bias.
6. In general, failures tend to be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ; however, they are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ in some subdomains.
7. In testing, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ are areas of the domain that are treated homogeneously by the software.
8. Errors tend to occur at the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of a subdomain.
9. A specific black-box testing approach is the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ method
10. The 6 steps in the category-partition method are: (1) identify \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (2) identify \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (3) partition \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (4) identify \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (5) produce \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (6) generate \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
11. When identifying categories in the category-partition method we are determining the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of each input element
12. When partitioning categories into choices we identify interesting \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of each category

Answer: Cases (subdomains)

1. For what two reasons do we identify constraints among choices?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. List the three types of constraints

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. When combining choices we use a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ constraint to make sure we don’t generate meaningless test cases
2. When combining choices we use a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ constraint to make sure a choice is tested only once because the choice is an error condition.
3. We generate test cases from \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

|  |
| --- |
| **Pay close attention to clip 20 (Category-Partition Demo). You will have a homework using this approach and using the TSL software.** |

**Omit the following two questions.**

1. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is an approach to model-based testing.
2. When developing test cases from a finite state machine you usually want to cover all the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_.