Final Term

Software Verification and validation

Marks: **50**

**Name: Ussama Zahid ID: 14895**

Q1. MCQS (10)

**1. When should company stop the testing of a particular software?**

**a.** After system testing done

**b.** It depends on the risks for the system being tested

**c.** After smoke testing done

**d.** None of the above

Ans**: b (It depends on the risks for the system being tested).**

**2. White-Box Testing is also known as \_\_\_\_\_\_\_\_ .**

**a.** Structural testing

**b.** Code-Based Testing

**c.** Clear box testing

**d.** All of the above

Ans: **d (All of the above).**

**3.  \_\_\_\_\_\_\_\_\_\_\_ refers to a different set of tasks ensures that the software that has been built is traceable to Customer Requirements.**

**a.** Verification

**b.** Requirement engineering

**c.** Validation

**d.** None of the above

Ans: **c (Validation).**

**4.  \_\_\_\_\_\_\_\_ verifies that all elements mesh properly and overall system functions/performance is achieved.**

**a.** Integration testing

**b.** Validation testing

**c.** Unit testing

**d.** System Testing

Ans: **d (System Testing).**

**5. What do you verify in White Box Testing?**  
*- Published on 03 Aug 15*

**a.** Testing of each statement, object and function on an individual basis.

**b.** Expected output.

**c.** The flow of specific inputs through the code.

**d.** All of the above.

Ans: **d (All of the above).**

**6. \_\_\_\_\_\_\_\_\_\_ refers to the set of tasks that ensures the software correctly implements a specific function.**  
*- Published on 03 Aug 15*

**a.** Verification

**b.** Validation

**c.** Modularity

**d.** None of the above.

Ans: **a (Verfication).**

**7. Who performs the Acceptance Testing?**  
*- Published on 03 Aug 15*

**a.** Software Developer

**b.** End users

**c.** Testing team

**d.** Systems engineers

Ans: **b (End users).**

**8. Which of the following is not a part of Performance Testing?**  
*- Published on 30 Jul 15*

**a.** Measuring Transaction Rate.

**b.** Measuring Response Time.

**c.** Measuring the LOC.

**d.** None of the above.

Ans: **c (Measuring the LOC).**

**9. Which of the following can be found using Static Testing Techniques?**  
*- Published on 29 Jul 15*

**a.** Defect

**b.** Failure

**c.** Both A & B

Ans: **a (Defect).**

**10. Testing of individual components by the developers are comes under which type of testing?**  
*- Published on 29 Jul 15*

**a.** Integration testing

**b.** Validation testing

**c.** Unit testing

**d.** None of the above.

Ans: **c (Unit testing).**

Q2**. Explain Black Box testing and White Box testing in detail. (10)**

Ans:

**White Box testing:**

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

**Working process of white box testing**:

* Input: Requirements, Functional specifications, design documents, source code.
* Processing: Performing risk analysis for guiding through the entire process.
* Proper test planning: Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
* Output: Preparing final report of the entire testing process.

**Testing Techniques names:**

1. Statement coverage.
2. Branch Coverage.
3. Multiple Condition Coverage.
4. Basis Path Testing.
5. Flow graph notation.
6. Cyclomatic Complexity.

**Block Box Testing:**

**Black box testing** is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications. In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

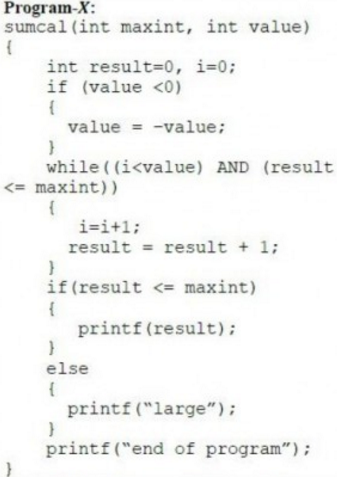
**Types of Black box testing:**

1. Functional testing.
2. Non-Functional testing.
3. Regression testing.

**Black box testing techniques:**

1. Equivalence class testing.
2. Boundary Value testing.
3. Decision table testing.

**Q3. Find the cyclomatic Complexity and draw the Graph of this code. (15)**



**Solution:**

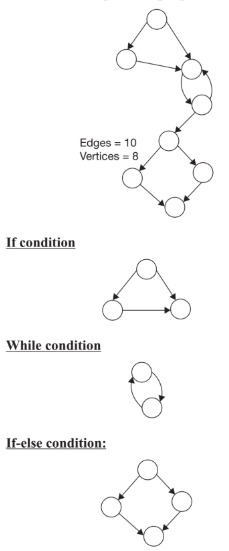
**Program X:**

Cyclomatic complexity of program x is the number of conditions+1

There are 2 **‘if’** conditions and 1 **‘while’** condition.

* Program ‘X’ = 4.

**Control flow graph:**



**Q4. What is Z specification and why its is used for, also give some example this code written in Z specification**. (15)

Ans:

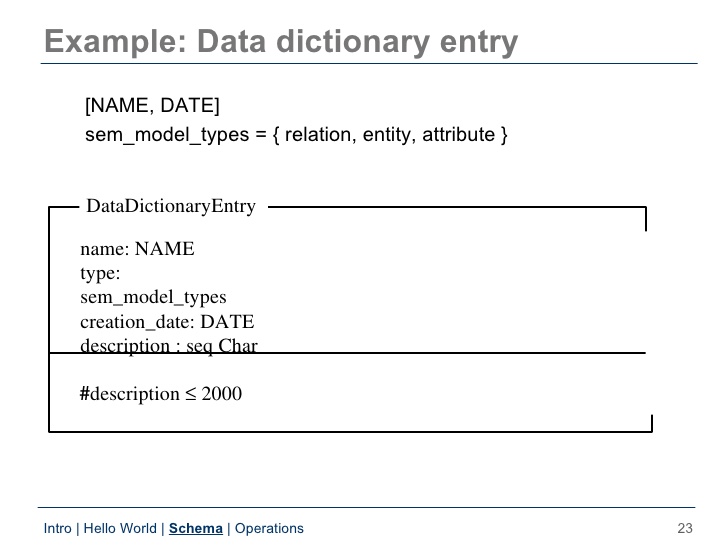
**Z specification:**

Z (pronounced zed) is a set of conventions for presenting mathematical text, chosen to make it convenient to use simple mathematics to describe computing systems. I say computing systems because Z has been used to model hardware as well as software.

Z is a model-based notation. In Z you usually model a system by representing its state -- a collection of state variables and their values -- and some operations that can change its state. A model that is characterized by the operations it describes is called an abstract data type (ADT). This modeling style is a good match to imperative, procedural programming languages that provide a rich collection of data types, and also to some physical systems (such as digital electronics) that include storage elements. Z is also a natural fit to object-oriented programming. Z state variables are like instance variables and the operations are like methods; Z even provides a kind of inheritance. However, Z is not limited just to ADT's and object-oriented style; you can also use Z in a functional style, among others.

**Used for:**

Z is based on the standard mathematical notation used in [axiomatic set theory](https://en.wikipedia.org/wiki/Axiomatic_set_theory), [lambda calculus](https://en.wikipedia.org/wiki/Lambda_calculus), and [first-order predicate logic](https://en.wikipedia.org/wiki/First-order_predicate_logic). All expressions in Z notation are [typed](https://en.wikipedia.org/wiki/Type_(model_theory)), thereby avoiding some of the [paradoxes of naive set theory](https://en.wikipedia.org/wiki/Naive_set_theory#Paradoxes). Z contains a standardized catalogue (called the mathematical toolkit) of commonly used mathematical functions and predicates, defined using Z itself



**Data dictionary modeling:**

* A data dictionary may be thought of as a mapping from a name “the key” to a value “the description in the dictionary”.
* Operations are:
  + **Add:** makes a new entry in dictionary or replaces an existing entry.
  + **Lookup:** give a name, returns the description.
  + **Delete:** deletes an entry from dictionary.
  + **Replace:** replaces the information associated with an entry.

Basic Data Representation



       DataDictionary


      ddict: NAME → DataDictionaryEntry




           Names         ...

Data dictionary initialization



     Init_DataDictionary

    ∆ DataDictionary


     ddict’ = φ




Intro | Hello World...Function Summary
Name                                    Symbol   dom f   One-to-   ran f
                                ...

Add and lookup operations

       Add_Error
      Ξ DataDictionary
      entry?: DataDictionaryEntry
      error!: seq cha...Add and lookup operations

       Add_OK
      ∆ DataDictionary                      Accessing sub
      entry?: DataDicti...