

**DEPARTMENT OF COMPUTER SCIENCE
MILITARY COLLEGE OF SIGNALS, NUST
SOFTWARE QUALITY ASSURANCE
BESE-14B**

Exam: Mid Term
Type of Paper: Regular
Semester: Fall

Instructor: Dr. Seemab Latif
Total Marks: 30
Time Allowed: 1.5 hours

Note:

1. This question paper has 2 pages and 4 questions.
 2. Attempt all questions.
 3. Use of programmable calculators with memory is not allowed
-

Question No.1:

(10)

Answer the following questions

- i. **What is Basis Path Testing?** A white box test case design technique that uses the algorithmic flow of the program to design tests.
- ii. **What is Beta Testing?** Testing of a rerelease of a software product conducted by customers.
- iii. **What is Boundary Testing?** Test which focus on the boundary or limit conditions of the software being tested. (Some of these tests are stress tests).
- iv. **What is Cyclomatic Complexity?** A measure of the logical complexity of an algorithm, used in white-box testing.
- v. **What is Quality Assurance?** All those planned or systematic actions necessary to provide adequate confidence that a product or service is of the type and quality needed and expected by the customer.
- vi. **What is Regression Testing?** Retesting a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made.
- vii. **How can it be known when to stop testing?** This can be difficult to determine. Many modern software applications are so complex, and run in such an interdependent environment, that complete testing can never be done. Common factors in deciding when to stop are:
 - Deadlines (release deadlines, testing deadlines, etc.)
 - Test cases completed with certain percentage passed
 - Test budget depleted
 - Coverage of code/functionality/requirements reaches a specified point
 - Bug rate falls below a certain level
 - Beta or alpha testing period ends
- viii. **What can be done if requirements are changing continuously?** A common problem and a major headache.
 - Work with the project's stakeholders early on to understand how requirements might change so that alternate test plans and strategies can be worked out in advance, if possible.
 - It's helpful if the application's initial design allows for some adaptability so that later changes do not require redoing the application from scratch.

- If the code is well-commented and well-documented this makes changes easier for the developers.
- Use rapid prototyping whenever possible to help customers feel sure of their requirements and minimize changes.
- The project's initial schedule should allow for some extra time commensurate with the possibility of changes.
- Try to move new requirements to a 'Phase 2' version of an application, while using the original requirements for the 'Phase 1' version.
- Negotiate to allow only easily-implemented new requirements into the project, while moving more difficult new requirements into future versions of the application.
- Be sure that customers and management understand the scheduling impacts, inherent risks, and costs of significant requirements changes. Then let management or the customers (not the developers or testers) decide if the changes are warranted - after all, that's their job.
- Balance the effort put into setting up automated testing with the expected effort required to re-do them to deal with changes.
- Try to design some flexibility into automated test scripts.
- Focus initial automated testing on application aspects that are most likely to remain unchanged.
- Devote appropriate effort to risk analysis of changes to minimize regression testing needs.
- Design some flexibility into test cases (this is not easily done; the best bet might be to minimize the detail in the test cases, or set up only higher-level generic-type test plans)
- Focus less on detailed test plans and test cases and more on ad hoc testing (with an understanding of the added risk that this entails).

ix. **What is the difference between quality assurance and testing?** Quality assurance involves the entire software development process and testing involves operation of a system or application to evaluate the results under certain conditions. QA is oriented to prevention and Testing is oriented to detection.

x. **What is the difference between verification and validation?**

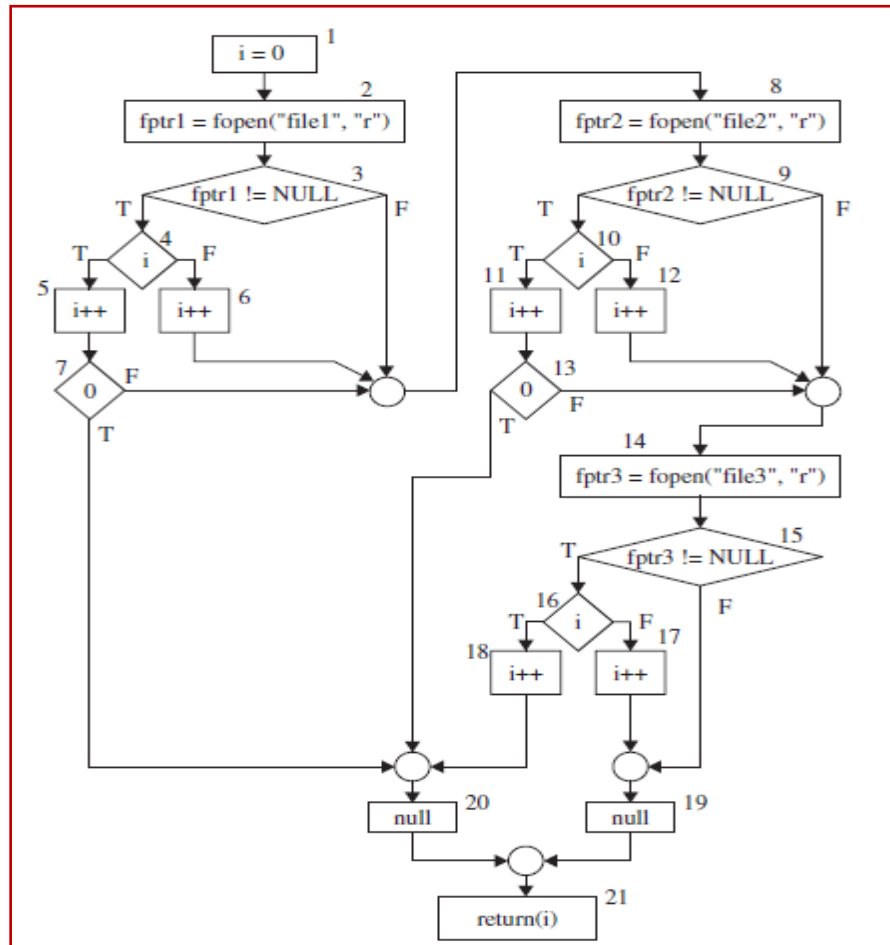
Question No.2:

(06)

Draw Control Flow Graph of the following piece of code using 100% multiple condition coverage. Also calculate its cyclomatic complexity.

```
FILE *fptr1, *fptr2, *fptr3; /* These are global variables. */
/* This function tries to open files "file1", "file2", and "file3" for read access,
and returns the number of files successfully opened. The file pointers of the
opened files are put in the global variables. */
```

```
int openfiles(){
    int i = 0;
    if(
        ((fptr1 = fopen("file1", "r")) != NULL) && (i++) && (0) ||
        ((fptr2 = fopen("file2", "r")) != NULL) && (i++) && (0) ||
        ((fptr3 = fopen("file3", "r")) != NULL) && (i++))
    )
    return(i);
}
```



Question No.3: (04)

By referring to the control flow graph obtained in Question 3, illustrate define-use-kill patterns for variables 'fptr2' and 'i' (taken in pairs as the paths are followed).

Question No.4: (04)

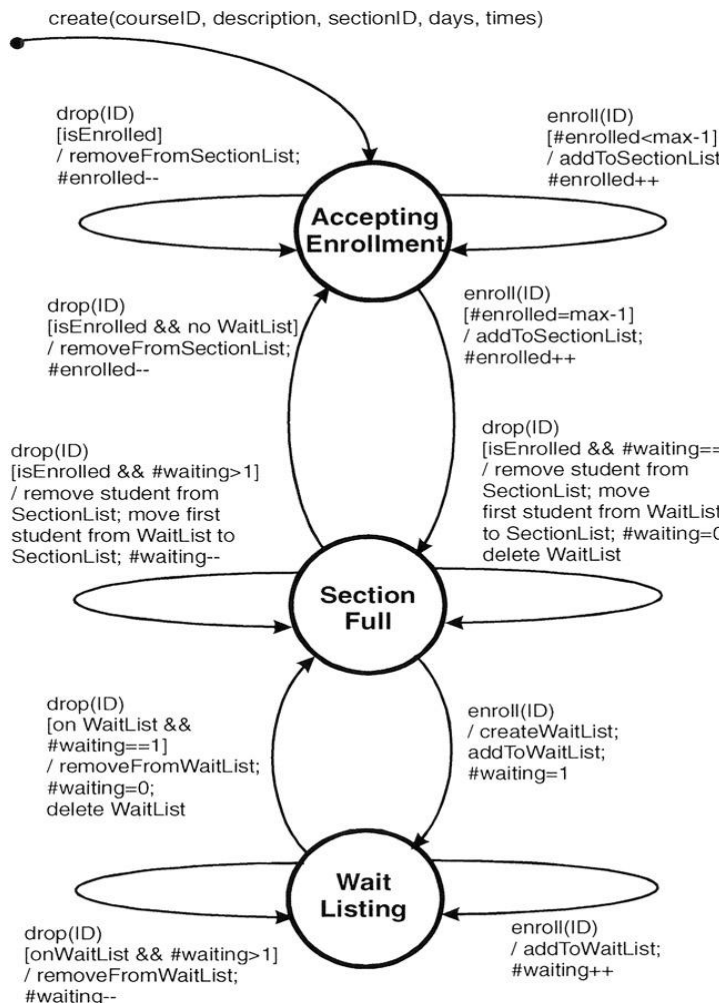
The top management was feeling that when there are any changes in the technology being used, development schedules etc, it was a waste of time to update the Test Plan. Instead, they were emphasizing that you should put your time into testing than working on the test plan. Your Project Manager asked for your opinion. You have argued that Test Plan is very important and you need to update your test plan from time to time. It is not a waste of time and testing

activities would be more effective when you have your plan clear. Use some metrics. How you would support your argument to have the test plan consistently updated all the time.

Question No.5:

(06)

Consider a University Registration System Web site. Below is a state-transition diagram for the "enroll in a course" and "drop a course" process. Determine a set of test cases that you feel adequately cover enroll and drop process.



Events

- create - Create a new course.
- enroll - Add a student to the course.
- drop - Drop a student from the course.

Attributes

- ID - The student identification number.
- max - The maximum number of students a course can hold.
- #enrolled - The number of students currently enrolled in the course.
- #waiting - The number of students currently on the Wait List for this course.

Tests

- isEnrolled - Answers "is the student enrolled (on the Section List)?"
- onWaitList - Answers "is the student on the WaitList?"

Lists

- SectionList - A list of students enrolled in the class.
- WaitList - A list of students waiting to be enrolled in a full class.

Symbols

- ++ Increment by 1.
- -- Decrement by 1.