Beijing Jiaotong University

2020—2021 School Year Second Semester Exam (A)

Course Name: Software Quality Assurance and Testing Techniques

Teacher: Xiaoping Che, Qian Zhang, Haiming Liu Major:_____

Class:_____ Name:_____ Student ID:_____

No.	1	2	3	4	5	6	Total Score
Score							
Examiner							

Part 1. Choose the correct answer from following choices. (2×10 marks)

1. Which of the following names is **not** a common name of a Bug?
   A. Fault
   B. Error
   C. Failure
   D. OMG

2. Which of the following testing method **cannot** be put in the same category with the others (in the perspective of verification and validation)?
   A. Acceptance Testing
   B. System Testing
   C. Integration Testing
   D. Unit Testing

3. Which of the following aspects does **not** belongs to the definition of Non-Functional aspects?
   A. Performance
   B. Correctness
   C. Usability
   D. Security

4. Which of the following is **not** included in the Specification attribute checklist?
   A. Completeness
   B. Testability
   C. Intuitive
   D. Code/Design-free

5. Which of the following is a correct description of testing?
   A. Testing is the process of demonstrating that errors are not present.
   B. The purpose of testing is to show that a program performs its intended functions correctly.
   C. Testing is the process of executing a program with the intent of finding errors.
   D. Testing is the process of establishing confidence that a program does what it is supposed to do.

6. Which testing method is **not** in the Control-flow Testing Criteria?
   A. Iteration Testing
   B. Statement Testing
   C. Branch Testing
   D. Path Testing

7. A Good UI design should **not** be?
   A. Consistent
   B. Comfortable
   C. Flexible
   D. Miscellaneous

8. In the code review part, which of the following error **cannot** be detected?
   A. Data Reference Errors
   B. Memory usage Errors
   C. Computation Errors
   D. Data Declaration Errors

9. Which of the following methods **cannot** be used to describe software specification?
   A. SDL
   B. GUI
   C. MSC
   D. Z NOTATION

10. Which of the following is **not** included in the Software Quality Assurance?
    A. SQP
    B. SQC
    C. SQV
    D. SQM

Part 2. The following descriptions are True of False? (2×5 marks)

11. It is possible to test a program completely. True or False

12. Testing can show the absence of bugs. True or False

13. Not all bugs found will be fixed. True or False

14. It is difficult to say when a bug is indeed a bug. True or False

15. In white box testing, "Exercise every statement at least once" equals to "Exercise every branch (in each direction) at least once" True or False

Part 3. Please fill the correct answer in the blanks. (2×5 marks)

16. In the Software Development V model, it contains Requirement Analysis, System Design, Program Design, Unit Testing, _____, System Testing, _____, and Operation Maintenance.

17. Please provide the full name of abbreviations SRS_____, PDL_____ in software testing domain.

18. There are two strategies in integration testing: _____ and _____.

19. In unit testing, _____ represents a component that calls component to be tested.

20. In Beta testing, rather than test team member, development team member, _____ will involve into the testing process.

Part 4. Short questions. (20 marks)

21. Please explain the definition of Equivalence Partitioning and Boundary Value Analysis, and their different impacts on designing test cases. (10 marks)

22. Please explain the definition of Validation and Verification, and explain the difference between them. (10 marks)

Part 5. Long Questions. (2*20 marks)

23. The schedule manager schedules a meeting to (a) time slot(s) as part of the 5-day schedule for the event. To accomplish this, the user must input three items:

1. The day (1, 2, 3, 4, or 5), chosen from a drop-down list.
2. The time slot(s), chosen through some type of user interface. Each day of the event lasts from 10am-6pm. There are 8, 1-hour time slots available per day (10-11am, 11am-12pm, 12-1pm, 1-2pm, 2-3pm, 3-4pm, 4-5pm, 5-6pm). A meeting can be scheduled for a 1-hour, 2-hour, or 3-hour time slot.
3. The name of the meeting (chosen from a drop-down list of meetings that the system already knows about)

The output would be a meeting being scheduled for a particular date and time (as reflected in the schedule). For example, if one of my test cases consisted of: {Day 3, 11am-1pm, The Go-Go's}, the expected output for this test case would be the Go-Go's being scheduled for Day3 of the festival in the 11am-1pm time slot.

(The minimum time unit is 1 hour. Minutes are not considered in this question)

Please provide the **Equivalence Partitioning** and **Boundary Value Analysis** of this use case and provide relevant **Test Cases Design**.

Test Case Example:

Test Case No.	Input Value 1	Input Value 2	Input Value 3	Expected Output	Description
1					

24. **(20 marks)** Now we have two code written by some programmers which are shown below, please provide the Control Flow Testing process of each code, including **Control Flow Graph, Cyclomatic complexity and Basis Set. And you can report the bugs if you find any.**

Code 1: (**8 marks**)

```
1. function radixSort(arr, maxDigit) {
2.         varmod = 10;
3.         vardev = 1;
4.         for(vari = 0; i < maxDigit; i++, dev *= 10, mod *= 10) {
5.                 for(varj = 0; j < arr.length; j++) {
6.                         varbucket = parseInt((arr[j] % mod) / dev);
7.                         if(counter[bucket]==null) {
8.                                 counter[bucket] = [];
9.                         }
10.                         counter[bucket].push(arr[j]);
11.                 }
12.                 varpos = 0;
13.                 for(varj = 0; j < counter.length; j++) {
14.                         varvalue =null;
15.                         if(counter[j]!=null) {
16.                                 while((value = counter[j].shift()) !=null) {
17.                                         arr[pos++] = value;
18.                                 }
19.                         }
20.                 }
21.         }
22.         return arr;
23. }
```

Code 2: (**12 marks**)

```
1.  public class TestBreakContinue {
2.  public static void main(String[] args) {
3.      int total = 0;
4.      System.out.println("Begin");
5.      while (true) {
6.          total++;
7.          int i = (int) Math.round(100 * Math.random());
8.          if (i == 88) {
9.              break;
10.         }
11.     }
12.     for (int i = 100; i < 150; i++) {
13.         if (i % 3 == 0) {
14.             continue;
15.         }
16.         System.out.println(i);
17.     }
18.     int count = 0;
19.     outer: for (int i = 101; i < 150; i++) {
20.         for (int j = 2; j < i / 2; j++) {
21.             if (i % j == 0)
22.                 continue outer;
23.         }
24.         System.out.print(i + "    ");
25.     }
26.  }
27. }
```