浙江大学 2016－2017 学年 fall-winter 学期

《Software Testing and Quality Assurance》

课程期末考试试卷

课程号：21120550，开课学院：_Computer Science

考试试卷：A 卷√、B 卷（请在选定项上打√）

考试形式：闭、开卷√（请在选定项上打√），

允许带_Textbook or Printed slides of lecture notes only, _入场

考试日期：2017 年 01 月 06 日 08:00~09:30　　考试时间：90 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：_____学号：_____所属院系：_____

题序	一	二	三	四	五	六	七	八	总 分
得分									
评卷人									

Answer three questions only, Question one is compulsory so that all students must attempt it. Answer the remaining two questions from any combination from Questions 2-4.

Question 1 [50 marks in total]

(a) A store in city offers different discounts depending on the purchases made by the individual. In order to test the software program that calculates the discounts, it is possible to identify the "price range" of purchase values that earn the different discounts. For example, if a purchase is in the price range of €1 up to and including €50 it has no discounts, a purchase in the price range of over €50 and up to €200 has a 5% discount, and purchases of price range €201 and up to €500 have a 10% discounts, and purchases of price range €501 to €2500 have 15% discounts. No items are sold outside these price ranges and if an incorrect price is entered the software returns a discount value of -1% to indicate that an error case has occured.

Considering the Black-Box Software testing method of Equivalence Partitions for this program write the input and output partitions. Then write out suitable tests ensuring to outline the partitions covered with each set of test data and highlight any error case with an asterisk. [15 marks]

(b) A program CheckEquilateral takes three integers as input that represents the lengths of the three sides of a triangle. The program first determines if the three numbers are valid inputs, that is, the length of any sides must not be 0 or a negative number. If not, it will return 'Not a triangle'. Otherwise, it will check whether it is an "Equilateral triangle" (three sides are the same length). Using the Causes and Effects as given below, complete the Truth Table underneath for the black box technique of Combinational Testing (note: impossible test cases have been eliminated). Following this, create suitable Test Data. [10 marks]

Causes	**Effects**
Side1>0	Not a triangle
Side2>0	Equilateral
Side3>0	Not Equilateral
Side1=Side2	
Side2=Side3	
Side1=Side3	

	Rules							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Causes								
Side1>0	F	*	*	T	T	T	T	T
Side2>0	*	F	*	T	T	T	T	T
Side3>0	*	*	F	T	T	T	T	T
Side1=Side2	*	*	*	F	F	F	T	T
Side2=Side3	*	*	*	F	F	T	F	T
Side1=Side3	*	*	*	F	T	F	F	T
Effects								
Not a Triangle								
Equilateral								
Not Equilateral								

(c) The software program Grade combines an exam and coursework mark into a single grade. The values for exam and coursework are integers. If the exam or coursework mark is less than 50% then the grade returned is a 'Fail'. To pass the course with a 'Pass, C', the student must score from 50% to 60% in the exam, and at least 50% in the coursework. They will pass the course with 'Pass, B', if they score more than 60% in the exam and 50% or more in the coursework. In addition to this, if the score of both the exam and the coursework is 90% or greater they are awarded a 'Pass, A'. Input values that are less than 0 or greater than 100 for either the exam or coursework are invalid and the program will return a message to say 'Marks out of range'.

Draw a Control Flow Graph (CFG) for the program Grade() using the source code given below. [10 marks] Then, derive test cases and test data for the following White Box Software testing methods: [15 marks]

i. Statement Testing

Line No.	**Code**
1)	`public static String Grade (int exam, int course) {`
2)	`String result="null";`
3)	`if ( (exam<0) || (exam>100) || (course<0) || (course>100) )`
4)	`result="Marks out of range";`
5)	`else {`
6)	`if ( (exam<50) || (course<50))`
7)	`result="Fail";`
8)	`else if (exam <= 60)`
9)	`result="Pass,C";`
10)	`else if ((exam>=90) && (course>=90))`
11)	`result="Pass,A";`
12)	`else`
13)	`result="Pass,B";`
14)	`}`
15)	`return result;`
16)	`}`

Only do two Questions from the following [25 marks each]:

Question 2

(a) Say why Exhaustive software testing is generally very difficult. Use an example to illustrate your answer. [7 marks]

(b) Describe two significant differences between Black-box testing and White-box testing. [8 marks]

(c) A useful Software testing technique that can be used to check the value or strength of a set of software test data is "Fault Insertion" or "Mutation testing". Offut suggested five categories of mutants that are sufficient for this type of testing. What are they? [10 marks]

Question 3

(a) Explain the approach to software development that is known as 'Big Bang'. Give two drawbacks to using it. [7 marks]

(b) The V-model of software development is similar to the Waterfall model but it is more test-oriented emphasizing verification and validation throughout the process. Give an outline of the V-model in the form of a diagram. There are four documents produced, each of which is associated with pairs of phases in the model. Name these documents and say how they are used in implementing the model. Give one advantage and one disadvantage of the model. [18 marks]

Question 4

(a) How does the Incremental development model incorporate Software testing? Use a diagram to illustrate your answer. Give one advantage and one disadvantage of this approach to Software development. [10 marks]

(b) Draw a diagram of the XP, Extreme Programming, software development process. Ensure to highlight in your diagram the points at which software testing is carried out. [9 marks]

(c) Explain one benefit and one disadvantage to the DevOps technique. [6 marks]