浙江大学 2017－2018 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：2018 年 1 月 10 日,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) The specification for a software program that computes the ticket cost of a concert at a particular venue is given below. The fee is based on the age of the attendee and their membership status. People outside the age range of 18 to 50 are not allowed to buy a ticket.

If the age of the member is from 18 to 35 (inclusive) the ticket cost is €40 if they are not a member and it is €35 if they are a member

If the age of the member is from 36 to 50 (inclusive) the ticket cost is €65 if they are not a member and it is €55 if they are a member.

The program input consists of 2 parameters: int age and boolean isMember. The program output is the int ticketCost value in Euros. Any invalid input returns a value of -€1.

Considering the Black-Box Software testing methods of Equivalence Partitions for this program fill in table with the input and output partitions. Then write out suitable tests ensuring to outline the partitions covered with each set of test data and highlight the error cases with an asterisk. [23 marks]

Test Cases	**Input Partitions**	**Range**
	Parameter - age	
EP1		
EP2		
EP3		
EP4		
	Parameter – isMember	
EP5		
EP6		
	Output Partitions	
	Parameter - ticketCost	
EP7		
EP8		
EP9		
EP10		
EP11		

Test Data

Test No.	**Test Cases Covered**	**Inputs**		**Expected Outputs**
		Age	*isMember*	*ticketCost*
1				
2				
3				
4				
5				
6				

(b) Using the Causes and Effects as given below, complete the Truth Table underneath for the black box technique of Combinational Testing. Then redraw the truth table to eliminate any impossible test cases. Following this, create suitable Test Data. [15 marks]

Causes	**Effects**
18<=age<=35	ticketCost=35
36<=age<=50	ticketCost=40
isMember	ticketCost=55
	ticketCost=65
	ticketCost=-1

First truth table

	Rules							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Causes								
18<=age<=35								
36<=age<=50								
isMember								
Effects								
ticketCost=35								
ticketCost=40								
ticketCost=55								
ticketCost=65								
ticketCost=-1								

Test Data

Test No.	**Test Cases Covered**	**Inputs**		**Expected Outputs**
		age	*isMember*	*ticketCost*
1	TT 1			
2	TT 2			
3	TT 3			
4	TT 4			
5	TT 5			
6	TT 6			

(c) This program ‘PhoneIns’ computes the cost of a smartphone insurance policy and outputs a value for the premium as denoted by *p*. It takes two inputs of integer *age* and Char *OS* (Operating System) type.

If the age entered is less than 16 or greater than 99 the program returns a premium of zero, *p*=0.
The input for OS takes the form of ‘I’ for IoS, ‘A’ for Android, and ‘W’ for Windows. If an incorrect value for the OS is entered, the program returns *p*=0.

In general the insurance premium is €50, *p*=50.

However, if a person has an iPhone and is under 25 then an extra €25 is added to the premium, *p*=75.

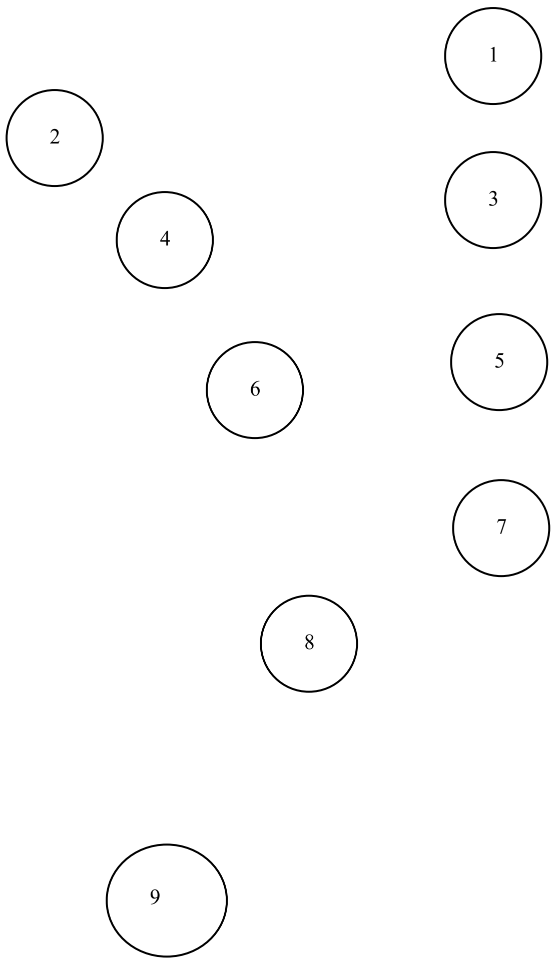
If the person is aged between 40 and 60 (inclusive) and they have an Android phone the premium falls by €10, *p*=40.

If the person is aged between 61 and 65 inclusive the premium falls by €5, *p*=45.

Line No.	Code
1	`public int phoneIns (int age, char OS) {`
2	`int p;`
3	`if ((age<16) || (age>99) || (OS!='I' &&OS!='A' &&   OS!='W'))`
4	`p=0;`
5	`else {`
6	`p=50;`
7	`if ((age<25) && (OS=='I'))`
8	`p += 25;`
9	`else {`
10	`if ((age>=40) && (age<=60) && OS=='A')`
11	`p -= 10;`
12	`else if ((age>=61) && (age<=65))`
13	`p -= 5;`
14	`}`
15	`}`
16	`return p;`
17	`}`

Complete the Control Flow Graph (CFG) for the program PhoneIns() using the source code provided. Then, derive test cases and test data for the White Box Software testing method of Statement Testing. Ensure to show the nodes covered by each set of test data [9 marks]

What would you need to add to the test data to satisfy Branch Coverage? Complete the table for Branch Testing, also showing the Branches covered by the test data [3 marks]



CFG

PhoneIns

Test No.	Test Cases Covered	Inputs		Expected Outputs
		Age	*OS*	*p*
1				
2				
3				
4				

Test No.	Test Cases Covered	Inputs		Expected Outputs
		Age	*OS*	*p*
1				
…	…	…	…	…

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

Question 2

(a) “The objective of software testing is all about finding software faults”. Can you list and explain 3 categories of software faults? [9 marks]

i. Documentation faults
ii. Stress or overload faults
iii. Throughput or performance faults
iv. Recovery faults
v. Syntax Faults

(b) Concisely describe three approaches to Debugging software. [3 marks]

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

(d) Briefly explain the difference between Unit and System testing. [5 marks]

Question 3

(a) Deciding when the testing of a piece of software should finish can be difficult to judge. Can you suggest two different criteria or benchmarks that could be useful to a Software Testing manager when making that decision? [5 marks]

(b) What is a control flow graph (CFGs)? Draw sample CFGs illustrating their appearance for a Switch statement and a While loop. [5 marks]

(c) A ‘data-flow’ approach to testing is a way of looking at a program as a flow of data from one statement to another. The motivation is to find data flow anomalies. Give explanations of three types of anomalies with short examples. [5 marks]

(d) Give one of the strengths and one of the weaknesses of *five* of the following White-box and Black-box software testing methods [10 marks]:

i. Statement Testing
ii. Branch Testing
iii. Path Testing
iv. Equivalence Partitioning
v. Boundary Value analysis
vi. Combinational Testing

Question 4

(a) State the difference between Software verification and Software validation in the context of software testing. [2 marks]

(b) Can you give *one* advantage and *one* disadvantage of using the V-model for software development? [4 marks]

(c) The Incremental development model of Software Development is an 'Agile' method. Draw a diagram that illustrates how the Incremental development model works. Give *one* advantage and *one* disadvantage of it. [7 marks]

(d) For the SCRUM approach to managing software development explain the following two terms. Also, sketch a diagram to show their relationship to each other. [10 marks]

   i. Product Backlog
   ii. Sprint and Sprint Backlog

(e) Explain one benefit to the DevOps technique. [2 marks]