UNIVERSITY OF MAURITIUS

FACULTY OF ENGINEERING

SPECIAL RETAKE EXAMINATIONS

AUGUST/SEPTEMBER 2016

PROGRAMME	**BSc (Hons) Information Communication and Technologies** **BSc (Hons) Electronics with Computer Science**		
MODULE NAME	**Software Engineering**		
DATE	**Saturday** **03 September 2016**	**MODULE CODE**	**CSE2001Y(5)**
TIME	**09:30 – 12:30 Hours**	**DURATION**	**3 Hours**
NO. OF QUESTIONS SET	**5**	**NO. OF QUESTIONS TO BE ATTEMPTED**	**5**

INSTRUCTIONS TO CANDIDATES

Answer All questions.

All questions carry equal marks.

Class Information Sheet is attached.

Question 1

(a) Implement a class named **Transaction** with the following **private** fields:

trans_id	<int>	id of transaction
amount	<float>	volume of transaction (in rupees)
disc_rate	*<float>*	discount rate

The class contains

- a constructor that requires as arguments the *volume of transaction* and automatically assigns the **transaction id** next in the sequence (1000 initially, 1001, 1002, …so on and so forth) and the discount rate according to the following table:

Volume of transaction (rupees)			
0 - 500000	500000 - 1000000	1000000 - 5000000	Above 5000000
0%	3%	4%	10%

- a method *Display()* that displays the transaction details

- a method *Cost()* that returns that the amount of to be paid after the discount is applied.

[3 + 2 + 4 Marks]

(b) Implement another class named **TransactionTransport**, which inherits from **Transaction**. A transaction with transport carries an additional charge based on the distance and on the vehicle required for the transport of the merchandise to the customer's place:

	Mini-truck	Lorry
Short	200	400
Average	400	700
Long	600	1000

TransactionTransport class will have two additional **private** fields:

vehicle_type	<String>	Type of transport, mini-truck or lorry
distance	*<String>*	How far customer site is

The class contains the following methods:

- a constructor that requires arguments for the *volume of transaction, the distance,* and the *type of vehicle* for the transport.

(continued next page)

(Question 1 continued)

- a method *Display()* that displays the details of the transaction with transport facilities.

- a method *Cost()* that returns that the amount of to be paid after the discount and transport charges are applied.

[3 + 3 + 4 Marks]

(c) Implement a class **CreateTransaction** that will create an array of 5 transaction instances as follows:

Transaction	**Volume**	**Transport Facilities**	**Distance**	**Type**
1	300000	N/A		
2	6500000		far	Lorry
3	5000	N/A		
4	8900000		short	Lorry
5	230000		short	Mini-truck

Your program should then display

- The details of each transaction
- The sum-total of charges of all transactions.

[2 + 4 Marks]

Question 2

(a) Consider the following relationships between components of a Java program:

(i) class B implements interface A
(ii) class B extends abstract class A

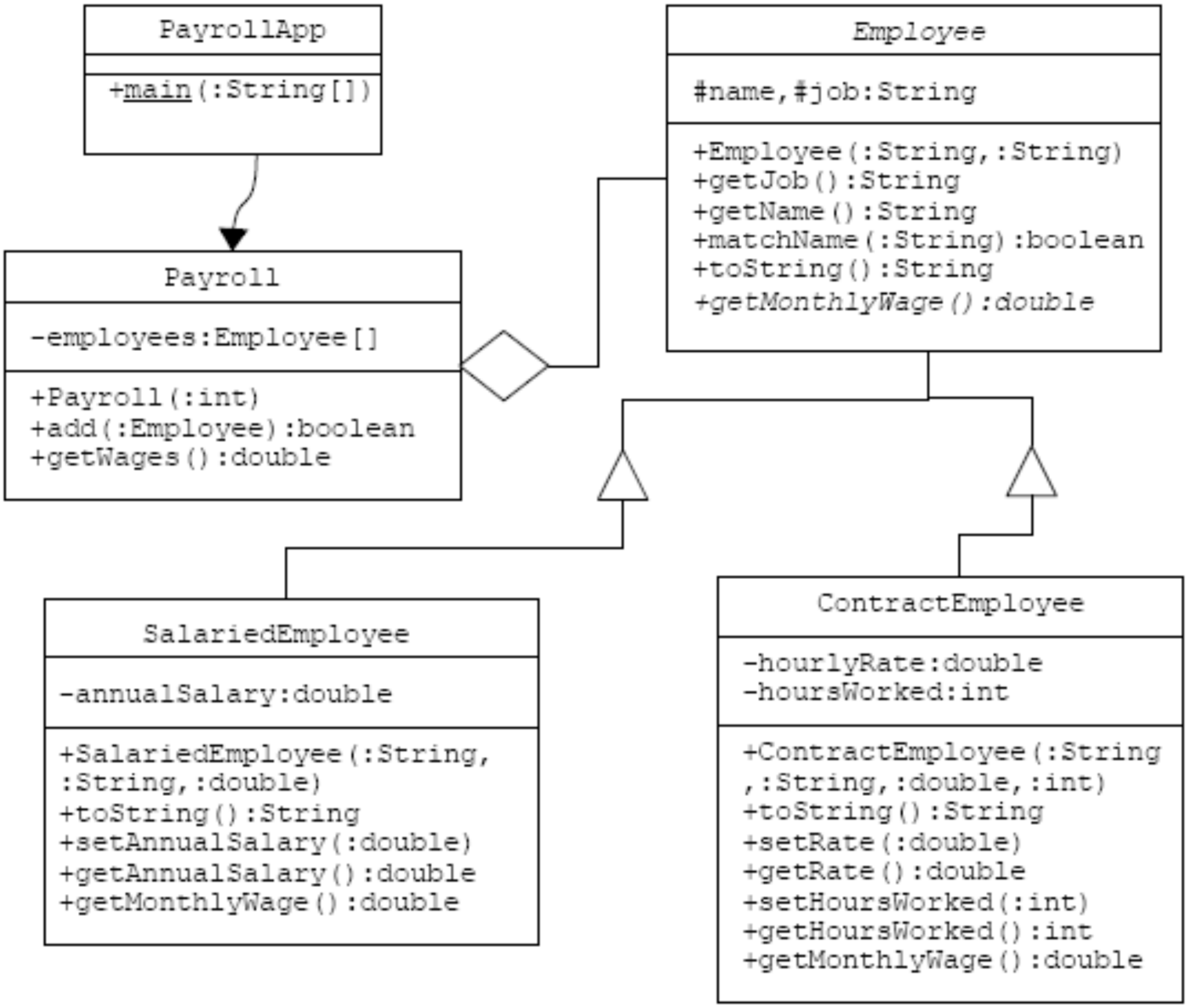
Briefly explain each of these relationships, stating any restrictions on, and significant implications of, its use. For each, include an example illustrating an appropriate use of the relationship.

[5 Marks]

(b) The UML class diagram below models employees in a company and the payroll that stores all employees.

(continued next page)

(Question 2 continued)



(continued next page)

(Question 2 continued)

Some method descriptions are given below:

Employee Class

```
+toString():String
//Compose and return a String from the attribute name and job

+matchName(:String):boolean
//See if the name of the current employee object matches another String.
```

SalariedEmployee Class

```
+toString():String
//Compose and return a String from the attribute name, job and annual salary

+getMonthlyWage():double
//returns the salary earned per month
```

ContractEmployee Class

```
-hoursWorked:int
//this private field indicates the number of hours worked in a month

+toString():String
//Compose and return a String from the attribute name, job, hourlyRate and hoursWorked

+getMonthlyWage():double
//returns the monthly salary based on the hourly rate and the number of hours worked
```

Payroll Class

```
+Payroll(:int)
//Create the array object given the size of the array, initially no employee

+add(:Employee):boolean
//Add a new employee to the payroll.

+getWages():double
//Get the total monthly wages of all employees in the payroll.
```

PayrollApp Class

```
+main(:String[])
//create a payroll of 5 elements
//create some employees and store them into payroll
//calculate and print total monthly wage
```

Study the above UML model and implement all the five classes.

[20 Marks]

Question 3

Read the following problem statement:

An Automated Ticketing System relies on computer processing and large databases to manage the distribution of tickets to entertainment and sporting events. Customers can purchase their tickets from many locations including the box office where the event will be held or any authorized retail locations. Online point-of-sale terminals are used to record the transaction and print the tickets, while at other locations the transaction is recorded and the customer receives the ticket by mail.

When a request is made, the system is designed to determine automatically from it master house calendar whether the performance requested by the user (event and date) is actually scheduled. The system may suggest alternate events on the same date or different dates of the same event. When the customer selects a date and event, the system shows seats that are available from its master seating chart. As seats are given, the master chart is updated.

(a) State four views that UML provides to model a system.

[2 Marks]

(b) Give a use-case diagram for the above system

[7 Marks]

(c) Give a class diagram for the above system.

[8 Marks]

(d) Read the following addendum for the above system:

At the terminal machine, the customer begins by inserting its bank credit card. If the card is valid, a list of different events is displayed. The customer then selects a particular event, specifies a seat or a number of seats from the seating chart, and confirms purchase of the ticket/s. If the transaction is approved, the seating chart is updated and the tickets are printed out and dispensed to the customer.

(i) List all the possible scenarios that may take place while purchasing a ticket from a point-of-sale terminal.

(ii) Draw one interaction diagram for the normal scenario.

[3 + 5 Marks]

Question 4

(a) In order to help us categorize software quality factors, McCall proposes a categorization which focuses on three important aspects of a software product namely Product Operation, Product Revision and Product Transition. Classify the following quality factors under those three categories.

- Interoperability: ability to interface with other components/systems
- Correctness: completeness of required output, availability of the information
- Testability: support for testing (e.g. log files, automatic diagnostics, etc)
- Efficiency: resources needed to perform software function
- Maintainability: effort to identify and fix software failures (modularity, documentation, etc)
- Portability: adaptation to other environments (HW/SW)

[6 Marks]

(b) In an information system the valid range of values for a particular field ‘amount’ is 10,000 to 50,000. Derive test cases for testing the field ‘amount’ using:

(i) Equivalence partitioning method
(ii) Boundary value analysis method

[4 Marks]

(c) Read the following pseudocode:

```
Read Length (l)
Read Breadth (b)
IF l > 4 THEN
       Print “invalid length”
ENDIF
IF b > 5 THEN
       Print “invalid breadth”
ENDIF
AREA= (l*b)
Print (“Area=”Area)
IF Area>12 THEN
       Print “False Area”
ELSE IF Area<=12 and Area>=1 THEN
       Print “True Area”
ELSE IF Area < 1
       Print “Area undefined”
```

(i) Draw the program flowchart.
(ii) Define cyclomatic complexity and state the cyclomatic complexity of the above program.
(iii) Identify the testcases for the above program.

[5 + 4 + 6 Marks]

Question 5

(a) Distinguish between software process and software product metrics. **(4 Marks)**

(b) At which stages of software development projects would you use the following software metrics and software quality ‘measures’:

(i) Function Points
(ii) Lines of Code Count (LOC)
(iii) Cyclomatic Complexity
(iv) Coupling and Cohesion

Explain why for each case.

[8 Marks]

(c) Discuss four key features or practices of Agile Development and explain the rationale for each. **[8 Marks]**

(d) Define Requirements Traceability and explain why it is relevant to the maintenance of software systems. **[5 Marks]**

END OF QUESTION PAPER

sg/