

REGENT UNIVERSITY
COLLEGE OF SCIENCE AND TECHNOLOGY



EXAMINATION PAPER

END OF SEMESTER EXAMINATIONS

DECEMBER 2013

**COURSE: OBJECT-ORIENTED PROGRAMMING
USING C++**

COURSE CODE: SICS 1523

LEVEL: 200 (Engineering)

TIME: 2½ HOURS

LECTURER: KENNETH K. AZUMAH

READ ALL INSTRUCTIONS VERY CAREFULLY

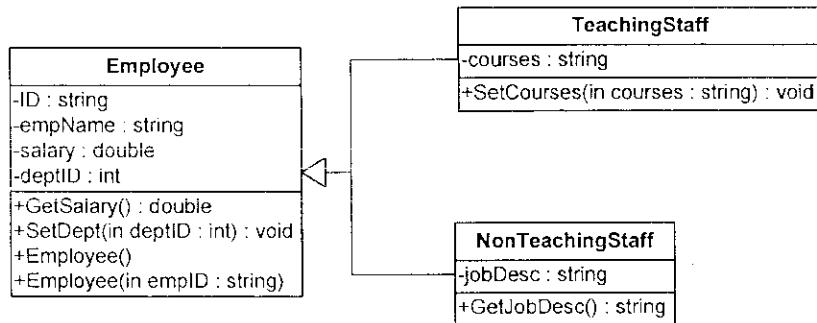
SECTION A

Answer any one question from this section

Question 1

[40 marks max]

Study the UML diagram below and answer the questions that follow:



a. What object-oriented property/properties is/are being portrayed?
[2 marks]

b. Translate the diagram directly into C++ code taking note of the following hints:

- Create three different and complete classes
- The employee class should have two constructors each initialising the attributes to some default values of your choice
- The accessors and mutators in each class return or set the values of the attributes contained in the function name

[28 marks]

c. Write a C++ main programme to add up two vectors **C** and **D** and print and print the result to the screen.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} + \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix}, \quad C = \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} \text{ and } D = \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix}$$

Hint: Use arrays in the main or create structure outside the main

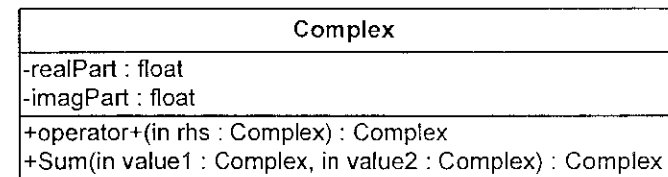
[10 marks]

Question 2

[40 marks max]

a. The UML diagram below show a **Complex** class with two attributes. Implement fully the class in C++ code taking note of the following hints:

- The operator '+' is overloaded to enable two **Complex** objects to be added (like **A + B**) [The first object is the class itself and the second on the input parameter]
- The *Sum* function takes two parameter objects and returns their sum
- In adding two **Complex** objects, sum their respective **realPart** and **imagPart** values



Hint:

```

Complex operator- (Complex rhs) {
    Complex result;
    Result.realPart = this->realPart + ...
    :
    :
}
    
```

[30 marks]

b. A programme is to be designed where any word entered is to be printed out in a reverse column format. Write out a C++ programme to demonstrate this mentioned functionality. Put comments in your code. An example input and output is as follows:

```

Enter your word: PROGRAMME
E
M
M
    
```

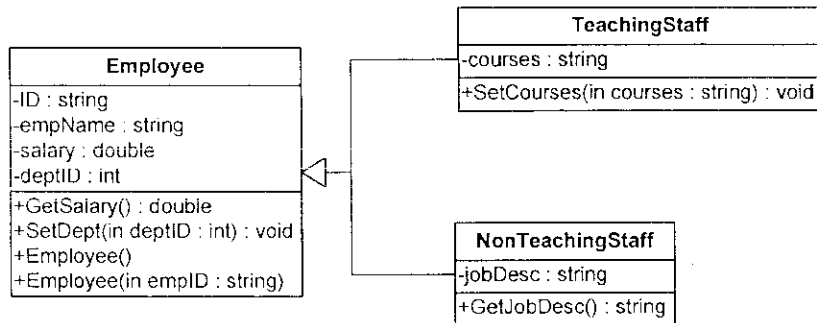
SECTION A

Answer any one question from this section

Question 1

[40 marks max]

Study the UML diagram below and answer the questions that follow:



a. What object-oriented property/properties is/are being portrayed?
[2 marks]

b. Translate the diagram directly into C++ code taking note of the following hints:

- Create three different and complete classes
- The employee class should have two constructors each initialising the attributes to some default values of your choice
- The accessors and mutators in each class return or set the values of the attributes contained in the function name

[28 marks]

c. Write a C++ main programme to add up two vectors **C** and **D** and print and print the result to the screen.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} + \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix}, \quad C = \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix} \text{ and } D = \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix}$$

Hint: Use arrays in the main or create structure outside the main

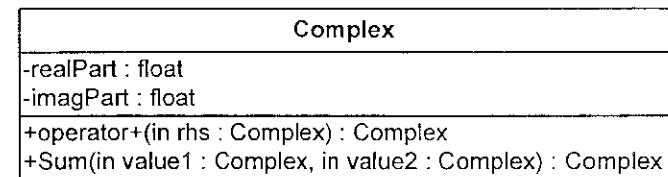
[10 marks]

Question 2

[40 marks max]

a. The UML diagram below show a **Complex** class with two attributes. Implement fully the class in C++ code taking note of the following hints:

- The operator '+' is overloaded to enable two **Complex** objects to be added (like **A + B**) [The first object is the class itself and the second on the input parameter]
- The *Sum* function takes two parameter objects and returns their sum
- In adding two **Complex** objects, sum their respective **realPart** and **imagPart** values



Hint:

```

Complex operator+ (Complex rhs) {
    Complex result;
    Result.realPart = this->realPart + ...
    :
    :
}
    
```

[30 marks]

b. A programme is to be designed where any word entered is to be printed out in a reverse column format. Write out a C++ programme to demonstrate this mentioned functionality. Put comments in your code. An example input and output is as follows:

```

Enter your word: PROGRAMME
E
M
M
    
```

[10 marks]

Question 3

[40 marks max]

Write a C++ programme to compute following statistics of a set of real numbers stored in an array. The programme should print the results to the screen.

- a. mean [6 marks]
- b. variance [6 marks]
- c. mode [18 marks]
- d. median [10 marks]

Hints:

$$\text{Mean} = \frac{\sum x}{N};$$

$$\text{Variance} = \frac{\sum (x - \text{mean})^2}{N};$$

Mode = the value that occurs the most;

Median = the value occurring in middle of the ordered set of values

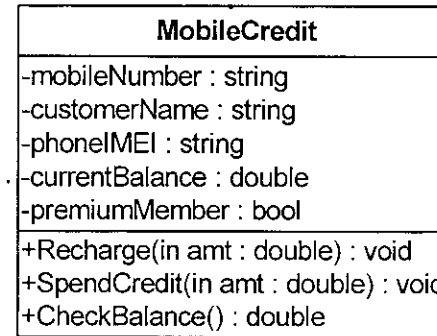
SECTION B

Answer any two questions from this section

Question 4

[30 marks]

a. Study the following UML class diagram named *MobileCredit*. Convert the class into C++ code implementing public accessor and mutator functions for each of the private attributes **except** currentBalance.



[16 marks]

b. Within the same class started in (a) above, implement the public member functions **Recharge**, **SpendCredit** and **CheckBalance** where the SpendCredit and Recharge functions decrease and increase the value of the currentBalance attribute respectively. [14 marks]

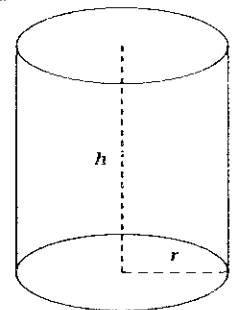
Question 5

[30 marks]

Consider the cylinder shown in the figure on the right and implement a class in C++ code along the following guidelines:

a. Determine the necessary attributes and their data types

[4 marks]



b. Create accessor and mutator functions for each of the attributes identified

[16 marks]

c. Create a parameterized constructor that will initialize all the attributes of the cylinder

[4 marks]

d. Add a default constructor (non-parameterized) that will initialize all the attributes of the cylinder

[6 marks]

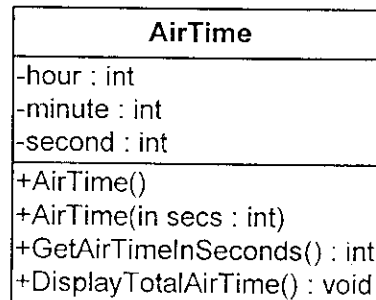
Question 6

[30 marks]

Examine the given UML diagram of a class named *AirTime* and answer the questions that follow (all answers may be included in one class definition):

a. Write the default constructor which initializes each attribute to 0 (zero).

[6 marks]



b. Implement the custom constructor that accepts time in seconds and initializes each attribute by converting the incoming time to hours, minutes and seconds.

[18 marks]

Hint: Hint: Use the % operator in the conversion

c. Implement the **DisplayTotalAirTime** member function that enables the class to display the Airtime attributes on the screen.

[6 marks]