MULTIMEDIA UNIVERSITY

STUDENT ID NO

SEAT NO: ____________________

MULTIMEDIA UNIVERSITY

SUPPLEMENTARY EXAMINATION

TRIMESTER 1, 2015/2016

TCP1201 – OBJECT-ORIENTED PROGRAMMING AND DATA STRUCTURES

(All sections / Groups)

19 NOV 2015
2.30 PM – 4.30 PM
(2 HOURS)

Question	Mark
1	
2	
3	
4	
Total	

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 12 pages with 4 Questions only.

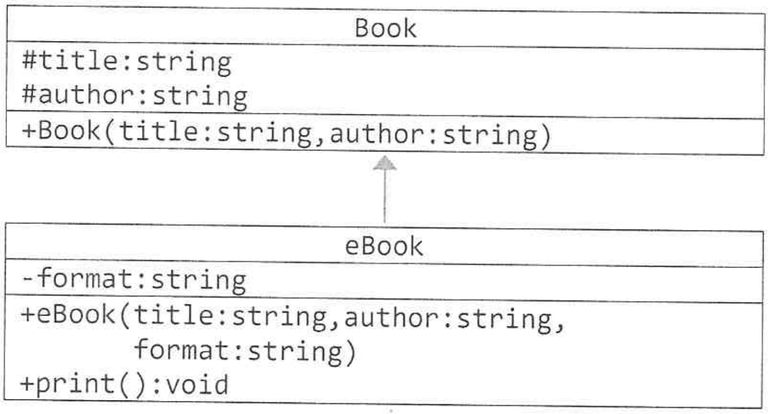
2. Attempt all **FOUR** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in **this Question Paper**.

Question 1

a. There are **three** basic steps when designing an object-oriented program. Write down the three steps. [2 marks]

b. The following UML Class Diagram is provided.



Answer the following questions:

i. Class Book and class eBook have a relationship. Write down the **name/type** of the relationship between the two classes. **Explain** your answer briefly. [1 mark]

Continued...

ii. A sample main function and its output are provided below. [5 marks]

```
int main () {
  eBook eb[3] = { eBook("OOP in C++","Lafore","pdf"),
                  eBook("Algorithms","Sedgewick","xps"),
                  eBook("DS in C++","Malik","mobi") };
  for (int i = 0; i < 3; i++)
    eb[i].print();
}
```

Sample output

```
OOP in C++, Lafore, pdf
Algorithms, Sedgewick, xps
DS in C++, Malik, mobi
```

Write a definition for the **eBook** class. Your definition should include a complete constructor and a complete print method.

Continued...

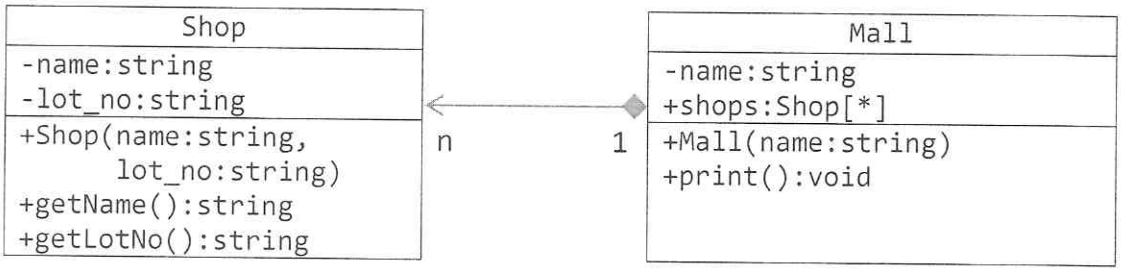
c. What makes a class **abstract**? Write a sample **definition** for a simple abstract class. What is the **limitation** or constraint of an abstract class when creating an object?

[2 marks]

Continued...

Question 2

a. The following UML Class Diagram is provided.



Answer the following questions:

i. Class Shop and class Mall have a relationship. Write down the **name/type** of the relationship between the two classes. **Explain** your answer briefly. [1 mark]

ii. A sample main function and its output are provided below. [4 marks]

```
int main() {
  Mall m("D'Pulze");
  m.shops.push_back (Shop("Subway", "SF08"));
  m.shops.push_back (Shop("Daiso", "UG11"));
  m.shops.push_back (Shop("Digi", "P09a"));
  m.print(); // produce the sample output below.
}
```

Sample output

```
Mall = D'Pulze
Shops:
Subway SF08
Daiso UG11
Digi P09a
```

Write a definition for the Mall class. Your definition should include a complete constructor and a complete print method. (Hint: Use vector)

Continued...

b. You need to overload operator '<<' and operator '==' for a class named **`Item`**. Write down the **function interfaces/prototypes** for overloading the two operators. (Function implementation is not required.)

[2 marks]

Continued…

c. Convert the following code segment to use the `try` and `catch` commands. The machine should be shutting down when its temperature exceeds 50 Celsius.

[3 marks]

```
double temperature;
cout << "Enter machine temperature: ";
cin >> temperature;

if (temperature > 50) {
  cout << "WARNING: Machine is too hot. ";
  cout << "Shutdown is activated.";
}
else
  cout << "Temperature is normal. Continue working.";
```

Continued...

Question 3

a. The definition of the **LinkedList** class is given below. Provide the implementation for the method **length** that returns the count of the number of elements in the linked list.

[3 marks]

```
template <typename T>
struct Node {
    T info;
    Node<T> *next;
};

template <typename T>
class LinkedList {
        Node<T>* start;
   public:
        LinkedList (): start(NULL) { }
        void insert (T &element );
        bool remove (T &element );
        int length();
        bool isEmpty();
};

template <typename T>
int LinkedList<T>::length () {





}
```

Continued...

b. (i) Which data structure, a Stack or a Queue is more suitable for implementing the **Back** button in a Web browser. Explain your answer.

[2 marks]

(ii) The definition of the **Queue** class is given below.

```
template <typename T>
struct Node {
    T info;
    Node<T> *next;
};

template <typename T>
class Queue {
        Node<T> *front;
        Node<T> *back;
  public:
        Queue(): front(NULL), back(NULL) { }
        void enqueue(T &element);
        bool dequeue(T &element);
        bool isEmpty();
};
```

Provide the implementation for the method **dequeue** function that removes the first element at the front of the queue.

[3 marks]

```
template <typename T>
bool Queue<T>::dequeue (T &element) {
```

Continued...

}

c. Consider the following pseudocode:

```
declare a stack of characters
while ( there are more characters in the word to read )
{
      read a character
      push the character on the stack
}
while ( the stack is not empty )
{
      write the stack's top character to the screen
      pop a character off the stack
}
```

What is written to the screen if the input word is "beautiful day"?

[2 marks]

Continued...

Question 4

a. Consider the following recursive function fun.

```
int fun(int x, int y)
{
        if (y == 0)   return 0;
        return (x + fun(x, y-1));
}
```

(i) Write the output of the function fun for the following input parameters.

[2 marks]

a) x = 9, y = 3:

b) x =5, y = 8:

(ii) Give the mathematical expression which shows what the function fun does.

[1 mark]

(b) Give the type of **STL associative container** that is most suitable for storage and retrieval of the name of lecturers based on the office room number. One office may accommodate one or two lecturers as illustrated in the mapping table below. **State two reasons for your choice.**

[3 marks]

Room No	Lecturer's name
1121	James Nelson
1122	Michelle Chan
1122	Carolyn Lee
1123	Ali Syaheel
1124	Rajeswari Kumaran

Continued...

(c) A **binary search tree (BST)** is a binary tree where each node has a comparable key and satisfies the restriction that the key in any node is larger than the keys in all nodes in that node's left subtree and smaller than the keys in all nodes in that node's right subtree. Complete the recursive function **insertInternal** to perform insertion of a new node to a BST.
Note: Each box may contain more than one line of code. [4 marks]

```
template<typename T>
structBSTNode {
  T info;
  BSTNode<T>* left;
  BSTNode<T>* right;
};

template<typename T>
classBinaryTree {
   BSTNode<T>* root;
public:
   ...
   bool insert (T &newElement) {
             return insertInternal (root, newElement);
   }
   bool searchInternal(BSTNode<T> *parent, T& newElement) {
        if (parent == NULL)   {



        } else if (                                  )



        else



        }
   }
   ...
};
```

End of Paper