MULTIMEDIA UNIVERSITY

STUDENT ID NO

SEAT NO: _____

SECTION: _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2018/2019

TCP1201 – OBJECT-ORIENTED PROGRAMMING AND DATA STRUCTURES

(All sections / Groups)

20 October 2018
2:30 p.m. – 4:30 p.m.
(2 Hours)

Question	Mark
1	
2	
3	
4	
Total	

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 11 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: **[25 Marks]**

a) List and provide a basic definition for the three most important characteristics of an Object Oriented programming Language. **[3 Marks]**

b) Draw a UML Class Diagram to represent the following elements from the problem domain of a football league. A football league is made up of at least four football teams. Each football team is composed of exactly twelve players, and one player captains the team. A team has a name and a record. Players have a number and a position. Football teams play games against each other. Each game has a score and a location. Teams are sometimes led by a coach. A coach has a level of accreditation and a number of years of experience and can coach multiple teams. Coaches and players are people, and people have names and addresses. **[8 Marks]**

Formulate a class diagram for this functional requirements and be sure to label all associations with appropriate multiplicities.

Continued ...

c) Write C++ code that will represent the following statements: **[10 Marks]**

Define a TEAM class that has the following attributes (name as a string, number of wins as an int, and number of losses as an int). These attributes cannot be accessed outside of the class. [2 Marks]	
Extend the TEAM class to have a constructor that accepts the name of a team, a number of wins, and a number of losses as arguments and sets the class properties to those values. [2 Marks]	
Overload the constructor of the TEAM class so that it takes only the name of the team as an argument. This constructor should set the name of the team to the argument passed by reference and then set the number of wins and losses to zeros. [1 Marks]	
Write a method [getName] that returns the name of the team. [1 Marks]	
Write a method [setName] that will set the team name to a new value passed as a reference. [1 Marks]	
Write a destructor for the class TEAM so that it will display a message saying **"Object is destroyed"**. [1 Marks]	
Write a main function to test the TEAM class. i. Declare a pointer variable [pt] that will point later to a TEAM object. [1 Marks] ii. Create a dynamic object of type TEAM and assign this object address to the pointer variable [pt]. This object's name is set to "MMU", the wins to 3, and the losses to 5. [1 Marks] iii. Change the name of the team pointed at by [pt] to "FCI". [1 Marks] iv. Print out the new name of the team object pointed at by [pt]. [1 Marks]	

Continued ...

d) What is the output generated by executing the following C++ main program? (assume all the required header files are included) **[4 Marks]**

```
using namespace std;
int main()
 {
    int x = 10, y = x%5;
    float z = 10;
    try {
        try {
            if (y == 0)
              throw string("divide by zero");
            z = x / y;
        } catch(string& e) {
            cout << "Called" << endl;
            throw runtime_error(e);
        }
    }
    catch(exception &e) {
        cout << "second catch "
                << e.what() << endl;
    }
    cout << z << endl;
   return 0;
}
```

Output:

Continued ...

Question 2: **[25 Marks]**

Given the following C++ class definition, answer (a) and (b):

```
class myPointer {
private:
 string label;
 int *data;

 void init (int sz, string
lbl)
{
  data = new int;
  *data = sz;
  label = lbl;
 }
public:
 myPointer()
 {
    init (2,"X");
 }
```

```
myPointer(int v,string lbl)
{
   init(v,lbl);
}

void set(int v, string lbl)
{
        *data = v;
        label = lbl;
}

void print()
{
  cout << label << " = "
       << *data << endl;
}
};
```

a) Explain why the main function shown below generates the [Generated output]. Extend the class [myPointerby] by adding one method to generate the output shown in the [Intended output].
You are not allowed to alter the main() function. **[8 Marks]**

```
int main()
{
    myPointer a;
    a.set(68,"Z");
    myPointer b(a);
    b.set(21,"W");
    a.print();
    b.print();
    return 0;
}
```

Generated output:
Z = 21
W = 21

Intended output:
Z = 68;
W = 21;

Continued ...

b) Add a member method to the class myPointer that will overload the addition [+] operator so that the program below will run and produce the output shown below.
The operator + is expected to concatenate the labels of the two objects and adds up their data. **[7 Marks]**

```
int main()
{
  myPointer a(10,"T");
  myPointer b(20,"M");
  myPointer c(a+b);
  a.print();
  b.print();
  c.print();
  return 0;
}
```

Output:

T = 10
M = 20
TM = 30

c) What is an abstract class in C++? **[4 Marks]**

Continued ...

d) Given the following class definitions: **[6 Marks]**

```
struct Point{
    int x;
    int y;
};

class Shape {
private:
    string name;
protected:
    Point  position;
public:
  void draw(){
   cout <<"Shape"
        << endl;
  }
};
```

```
class Triangle: public Shape{
public:
  void draw(){
   cout << "Triangle" << endl;
  }
};

class Rectangle: public Shape
{
public:
  void draw(){
    cout << "Rectangle"
         << endl;
    }
};
```

Compiling the main function causes two compilation errors. These errors are caused by the line (pr = ps;). You are expected to correct the line and overcome the compilation errors so that the program will generate the output shown below.

```
int main() {
    Shape *ps;
    Triangle *pt;
    Rectangle *pr;
    Triangle t;
    Rectangle r;
    ps = &t;
    ps->draw();
    ps = &r;
    ps->draw();
    pr = ps;    //---Error
    ps->draw();
    return 0;
}
```

Intended Output:

Triangle
Rectangle
Rectangle

Continued ...

Question 3: **[25 Marks]**

Given the following Linked List class specification, answer (a), (b) and (c):

```
template<class T>
class Node {
public:
    T info;
    Node<T> *next;
};
```

```
template<class T>
class LinkedList
{
private:
    Node<T> *head;
    Node<T> *tail;

public:
    LinkedList();
    LinkedList(T value);
    T back();
    T front();
    void push_back(T value);
    bool pop_back();
    void push_front(T value);
    bool pop_front();
    int size();
    bool empty();
    void print();
    ~LinkedList();
};
```

a) Construct a basic UML class diagram for the class specifications above. The UML diagram must show only names of classes, relationships, and multiplicities. **[4 Marks]**

Continued ...

b) Develop an implementation for the method push_back(T v) so that it adds a node to the end of the linked list. Your method must handle both cases of an empty list and a non-empty list. **[6 Marks]**

c) Given the following main method: **[6 Marks]**

```
int main()  {
    LinkedList<int> lst;
    lst.push_back(10);
    lst.push_back(12);
    lst.push_back(15);
    cout << lst.sum() << endl;
    return 0;
}
```

Write an appropriate implementation for a method **sum( )** that returns the sum of all of the elements in a linked list. The method must be defined as a template method.

Continued ...

d) Explain what the following program does and what will be the output generated when executing it? (Assume all the required header files are included) **[6 Marks]**

```
using namespace std;
int main() {
  stack<int> stk;
  for (int i=0;i<5;i++)
    stk.push(i);

  deque<int> que;
  while (!stk.empty()) {
   que.push_back(stk.top());
   stk.pop();
  }
  auto it=que.begin();
  while(it != que.end()) {
    cout << que.front();
    que.pop_front();
    it++;
  }
  return 0;
}
```

Explanation:

Output:

e) Given the following C++ program, write 3 C++ lines to rearrange the contents of the integer vector [v] in ascending order using an appropriate STL algorithm, then print out the content of the vector v on one line using range loop. **[5 Marks]**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    vector<int> v{1,6,5,4,8,7,9,3};









  return 0;
}
```

Output:
1, 3, 4, 5, 6, 7, 8, 9,

Continued ...

Question 4: **[25 Marks]**

a) Insert the following list of numbers into a binary search tree. Start the insertion of the numbers from left to right and in order. **[6 Marks]**

40 , 30 , 20, 10, 21 , 32 , 31 , 45 , 60 , 42 , 41 , 43

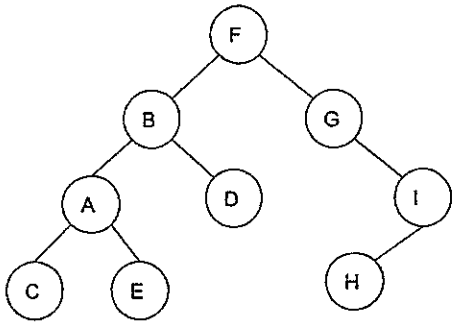
b) Using the BST in (a) identify and list the following: **[5 Marks]**

1. All the leaf nodes

2. All the nodes at level 1.

3. What is the total number of nodes in a binary tree at level n.

c) Give the results of traversing the following Binary Tree using: **[6 Marks]**



In-order Traversal	Pre Order Traversal	Post Order Traversal

Continued ...

d) Use the following classes' specifications: **[5 Marks]**

```
template<class T>
class Node {
public:
    T info;
    Node *left;
    Node *right;

    Node() {
      left = nullptr;
      right = nullptr;
    }
    Node(T v) {
      info = v;
      left = nullptr;
      right = nullptr;
    }
};
```

```
template <class T>
class BinaryTree {
    Node<T> *root;
public:
    BinaryTree();
    BinaryTree(T  value);
    :
};
```

Write a recursive method to print out the elements of a binary search tree while traversing its elements using preorder traversal.

e) Defend the idea of using a vector data structure to store a telephone directory over a using a list [linked list]. **[3 Marks]**

End of Paper.