

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

SEAT NO: _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TCP1201 – OBJECT-ORIENTED PROGRAMMING AND DATA STRUCTURES (All sections / Groups)

11 OCTOBER 2016
09:00 a.m. – 11:00 a.m.
(2 Hours)

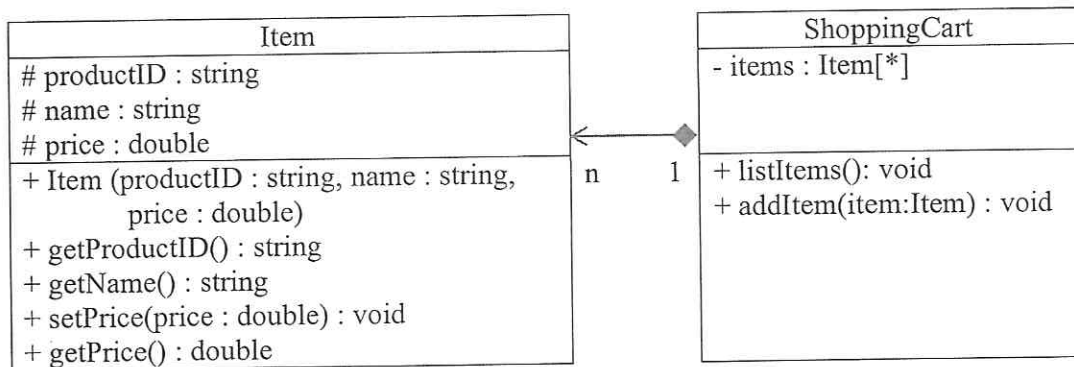
Question	Mark
Total	

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 13 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. The following UML Class Diagram is provided.



- i. Write the **class interface** (class declaration) for the two classes in the UML Class Diagram. Initialize the constructor methods with the **constructor initializer list**.
[3 marks]

Continued...

- ii. Name and describe briefly the relationship between the **ShoppingCart** class and the **Item** class in the UML diagram above.

[1 mark]

- iii. Provide the implementation code for the **setPrice** method.

[1 mark]

- iv. Provide the implementation code for the **listItems** method.

[2 marks]

Continued...

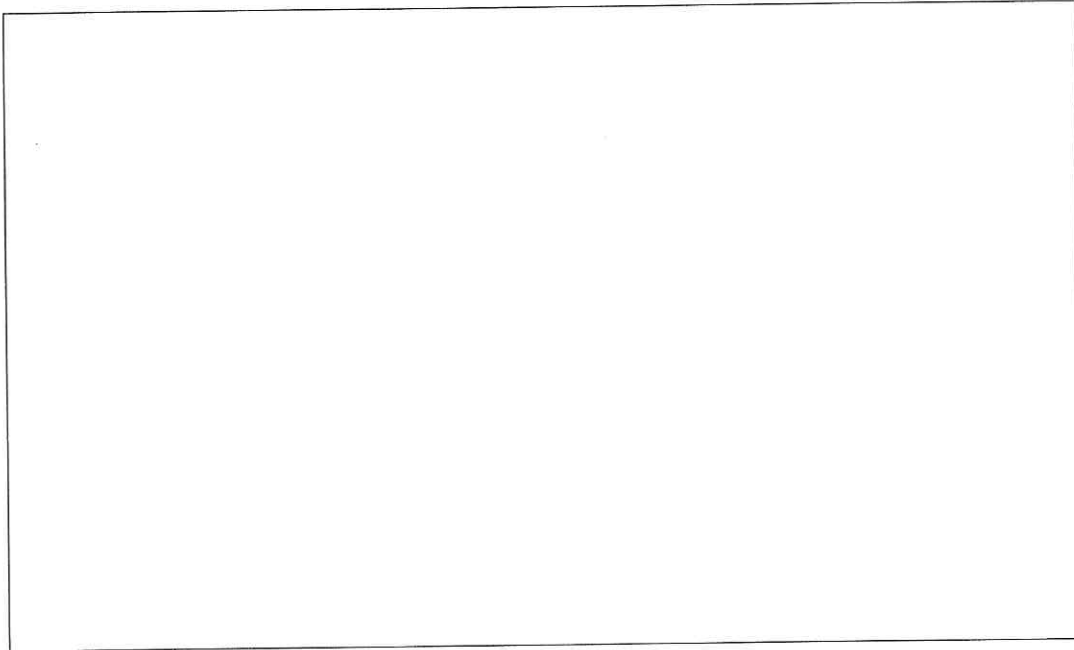
- v. Write a driver (main) program that creates an **instance** of the **ShoppingCart** class and add two items to the **ShoppingCart** instance to produce the following output:

Expected Output:

Shopping Cart:

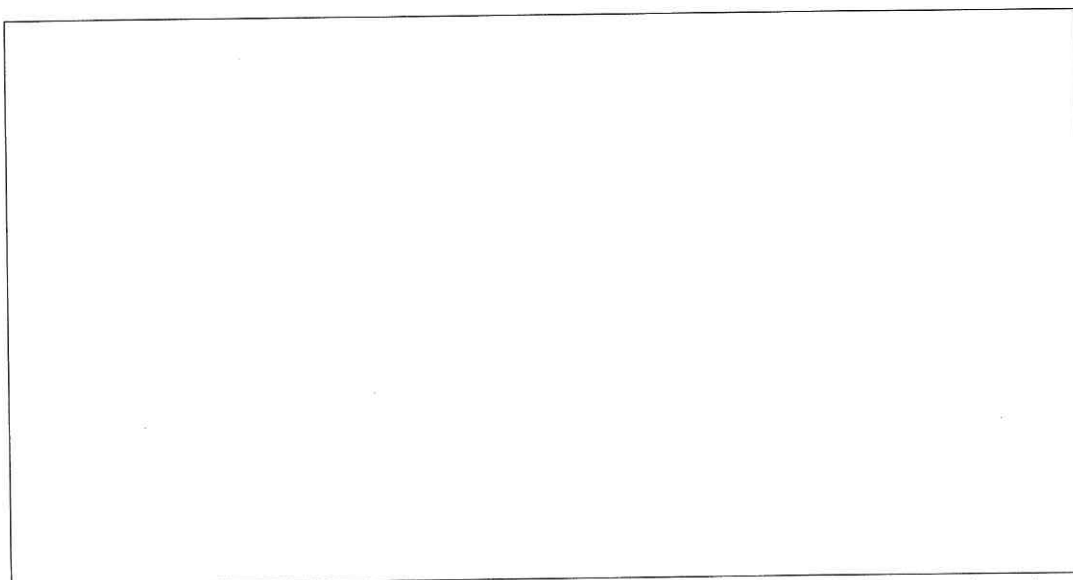
A8965	Sleeping Bag	48.00
C1361	Tent	90.00

[2 marks]



- b) Briefly describe one way of implementing **encapsulation** in an object-oriented program.

[1 mark]



Continued...

Question 2

a. Answer the following questions based on the program given below:

```
class Vehicle {  
    protected:  
        string manufacturer;  
    public:  
        Vehicle (string manufacturer): manufacturer (manufacturer) {}  
        virtual void print() = 0;  
};
```

```
class PassengerVehicle : public Vehicle {  
    protected:  
        int noOfPassengers;  
    public:  
        PassengerVehicle (string manufacturer, int noOfPassengers)  
            : Vehicle(manufacturer), noOfPassengers(noOfPassengers) {}  
};
```

i. **PassengerVehicle** is an abstract class. Explain briefly why it is an abstract class and how you can modify the class to make it a concrete class (without providing any code).

[2 marks]

ii. Provide the class definition of a new concrete class, **Bike** that is a sub class of **PassengerVehicle** and has an attribute **saddleHeight**. The class definition should include a constructor method to initialize all attributes, including attributes of the super classes. Besides, it should override the **print** function to print the values of all the attributes.

[2 marks]

Continued...



b. Consider the following program:

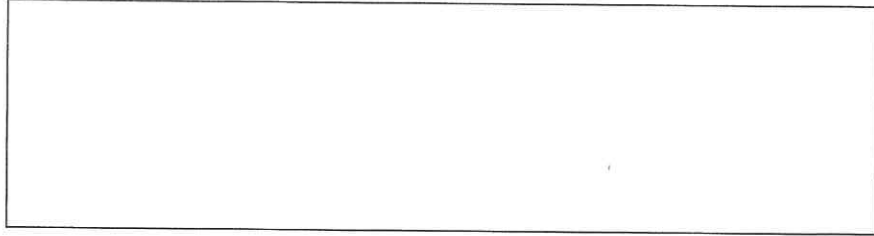
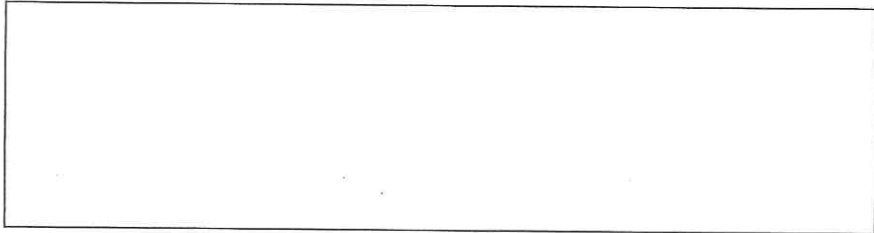
```
int main() {  
    int x;  
    cin >> x;  
  
    try {  
        if (x < 0)        throw x;  
        if (x == 0)       throw "zero";  
        cout << "x = " << x << endl;  
    }  
    }  
    catch (int a) {  
        cout << "Invalid input!" endl;  
    }  
    catch (...) {  
        cout << "Have a great day!" endl;  
    }  
  
    return 0;  
}
```

Write the output of the above program for the following input of:

[3 marks]

i. x = 0

Continued...

ii. $x = 99$ iii. $x = -8$ 

c. The following main program generates some run-time errors:

```
class Calculator {
public:
    int multiply(int x, int y) {
        return x*y;
    }
    int add(int x, int y) {
        return x+y;
    }
};

int main() {
    Calculator calc1;                                     //LINE 1
    int a = 10, b = 5;                                    //LINE 2
    cout << a << "+" << b << "=" << calc1.add(a,b) << endl;    //LINE 3
    cout << a << "x" << b << "=" << calc1.multiply(a,b) << endl; //LINE 4

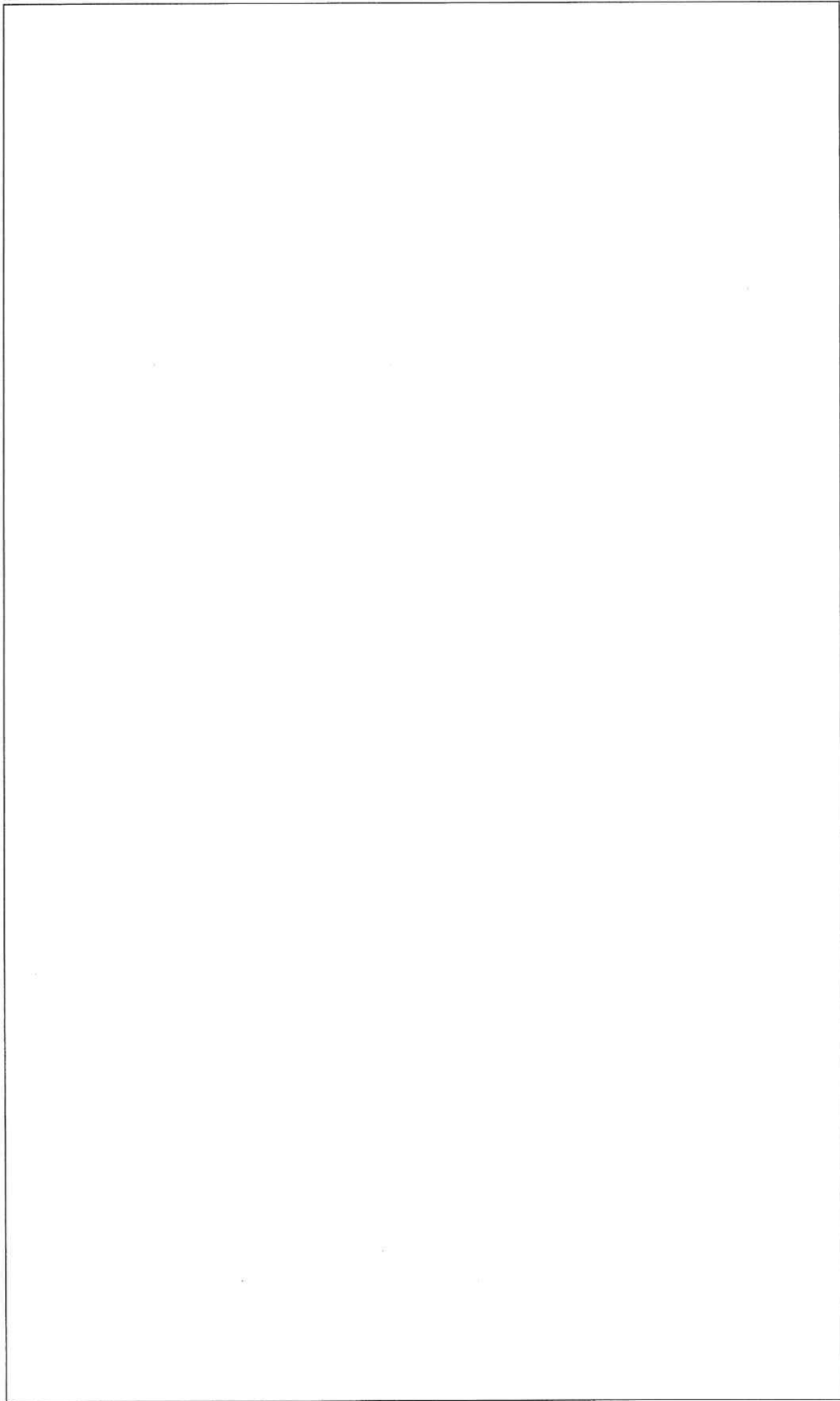
    Calculator calc2;                                     //LINE 5
    double p = 2.5, q = 1.3;                             //LINE 6
    cout << p << "+" << q << "=" << calc2.add(p,q) << endl;    //LINE 7
    cout << p << "x" << q << "=" << calc2.multiply(p,q) << endl; //LINE 8

    return 0;
}
```

Convert the **Calculator class** above to a template class. In order for the **main function** to work for the new template class, there are two lines of code that need to be modified. Specify the **line numbers** for these two lines of code and provide the **correct implementation** for each.

[3 marks]

Continued...



Continued...

Question 3

- a. Between Stack and Queue, which one is suitable to be used for implementing an **Undo** button in a word document? Explain your answer.

[2 marks]

- b. The following shows the definition of the **Queue** class. Provide the implementation for the **enqueue** method; the method that inserts an element at the end of the queue.

[4 marks]

```
template <typename U>
class Node {
    U item;
    Node<U>* next;
public:
    Node(const U& anItem);
    void setNext(Node<U>* nextNode);
    U getItem() const ;
    Node<U>* getNext() const ;
};

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

Continued...

```
template <typename U>
void Queue<U>::enqueue(U& newEntry)
{
```

```
}
```

- c. You need to implement a list to store sales data. The list allows you to insert, update, and delete data wherever you want. Do you think array-based or link-based list implementation is more appropriate? Explain your answer.

[2 marks]

Continued...

- d. Given below is the definition of the **LinkedList** class. Provide the implementation for the method **printList** that is used to print all items in the linked list. [2 marks]

```
template <typename T>
class Node {
    T item;
    Node<T>* next;
public:
    Node(const T& anItem);
    void setNext(Node<T>* nextNode);
    T getItem() const ;
    Node<T>* getNext() const ;
};

template <typename T>
class LinkedList {
    Node* headPtr;
public:
    LinkedList(): headPtr(nullptr){ }
    void insert(T &newEntry);
    bool remove();
    void printList();
    bool isEmpty();
};

template<typename T>
void LinkedList<T>::printList()
{
```

```
}
```

Continued...

Question 4

- a. The following is a recursive function **CF** with two input parameters, i.e. x and y .

```
int CF(int x, int y){  
    if(y==0)  
        return x;  
    else  
        return CF(y, x % y);  
}
```

Write the output of the function for the following input parameters.

[3 marks]

$x = 20, y = 15;$

$x = 300, y = 500;$

- b. Construct a binary search tree by inserting the following numbers one after another.

[2 marks]

6, 4, 8, 11, 2, 5, 3

Continued...

- c. Is the resulted binary search tree in part (b) a complete binary tree? Explain your answer.

[1 mark]

- d. The following is an incomplete program. It is a program that accepts 8 integers from a user and then displays them in ascending order. Use the STL container, i.e. multiset to implement this. Below are the sample run of the program.

[4 marks]

Sample run:

Enter 8 numbers: 8 3 5 6 8 11 2 3

Sorted numbers: 2 3 3 5 6 8 8 11

```
#include <iostream>
#include <set>
using namespace std;
```

```
int main(){
    multiset<int> mset;
```

```
    it<<" ";
```

```
    //0.5m
```

```
    return 0;
}
```

End of paper