BSc (Hons) in Information Technology
Year 1

Lab Week 13

IT1050 – Object Oriented Concepts Semester 1, 2023

Objectives:

- Create classes that have Aggregation and Composition Relationships

For this week's lab we will represent Batches (Groups), Students and Exams. A starter code with several classes will be provided to complete the lab work.

Exercise 1:

a) Use the Student.h and Student.cpp and create a main program

```
#include "Studdent.h"
int main() {
        Student *st1 = new Student("IT16231123", "Nimal");
        Student *st2 = new Student("IT16232332", "Surangi");
        st1->display();
        st1->displayLine();
        st2->display();
        st2->displayLine();
        return 0;
}
```

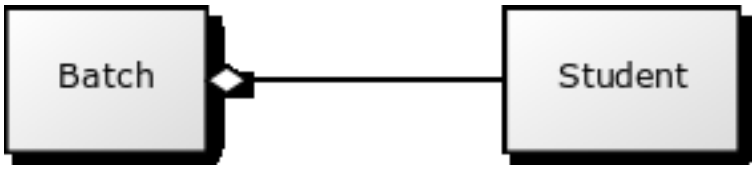
Exercise 2:

a) Use the Batch.h and Batch.cpp programs and add create Batch objects as suggested below in the main program.

```
Batch *batch1 = new Batch(1, "Year 1 Batch 1 - Malabe Campus");
Batch *batch2 = new Batch(1, "Year 1 Batch 2 - Malabe Campus");
batch1->display();
batch2->display();
```

Exercise 3:

Here we will implement the following aggregation relationship.



We will store Student objects as properties in the Batch Class.

(a) Modify the Batch.h and Batch.cpp classes as indicated below.

```
#ifndef _BATCH
#define _BATCH

#include "Student.h"
#include <string>
using namespace std;

#define MAXSTUDENTS 10

class Batch {
  private:
    int year;
    string name;
    int count;
  public:
    Student *students[MAXSTUDENTS];
    Batch(int pyear, string pname);
    void display();
    string getName();
};

#endif
```

(b) In the main program assign student objects to the Batch.

```
batch1->Students[0] = st1;
batch1->Students[1] = st2;
```

Display details of Student1 and Student 2 of Batch 1

```
batch1->Students[0]->display();
batch1->Students[1]->display();
```

Exercise 4:

(a) Make students[] to be a private property. This requires us to develop a method to add the Batch class to display student objects.

- In the Batch class move the property Student *students[MAXSTUDENTS] to be a private property.
- Add a method called void addStudent(Student *pstudent) in the Batch class to add a Student object.
- Hint use a count variable to keep track of the student objects stored in the Student arrays.

(b) Modify the Batch display() method to display student details as well.

(c) Add code in the main function to display details of Batch1 and Batch2.

Exercise 6:

(a) Modify the Student class to have a property that links to the Batch class

Add the myBatch property as a private property to the Student Class. (i.e. modify Student.h)

Batch *myBatch;

Add a method called void assignBatch(Batch *batch) in the Student class. In the code assign the batch detail to attribute mybatch.

(b) Since the Batch class uses the Student class and the Student class uses the Batch class in C++ we need to forward declare the classes. See the code below.

In Batch.h

```
9   #define MAXSTUDENTS 10
10  // Forward Declaration needed as both Student and Batch are refering each other
11  class Student;

12

13  class Batch {
14    private:
15       int year;
16       string name;
```

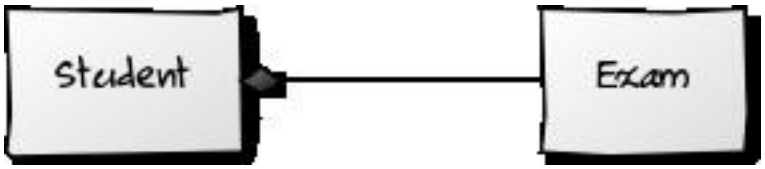
In Student.h

```
10  // Forward Declaration needed as both Student and Batch are refering each other
11  class Batch;

12

13  class Student {
14    private:
15       string name;
16       string studentID;
17       float GPA;
18       Batch *myBatch;
19    public:
20       Student();
```

Modify addStudent() method in the Batch class to call the Student assignBatch() method.

Exercise 7:

Use the Exam.h and Exam.cpp file to add details of exams to a Student.



We will model the composition relationship.

(a) Add Exam.h and Exam.cpp to your project.
(b) Add the property Exams *myExams[MAXEXAMS] to the Student.h file
(c) Implement the addExam(string module, int marks) to the Student class

i.e. You will need to add a property called examCount in the student class and set it to zero in the constructor. Also uncomment the code in the Student.cpp class displayResults()

(d) Modify the main program as follows.

```
1 int main() {
2    Batch *batch1 = new Batch(1, "Batch 1 - Malabe Campus");
3    Batch *batch2 = new Batch(2, "Batch 2 - Malabe Campus");
4
5    Student *st1= new Student("IT16231123", "Nimal");
6    Student *st2 = new Student("IT16232332", "Surangi");
7    Student *st3  = new Student("IT16123411", "Kamala");
8    Student *st4 = new Student("IT16327822", "Kaushi");
9
10
11   batch1->addStudent(st1);
12   batch1->addStudent(st2);
13   batch2->addStudent(st3);
14   batch2->addStudent(st4);
15   cout << "Display one Student's Details" << endl;
16   st4->display();
17
18   cout << "Display a Batch's Details " << endl;
19   batch1->display();
20   cout << "Display a Batch's Details " << endl;
21   batch2->display();
22
23   st1->addExam("OOC", 89);
24   st1->addExam("IWT", 67);
25   st1->addExam("SPM", 44);
26
27   st1->displayResults();
28
29   delete st1;
30   delete st2;
31   delete st3;
32   delete st4;
33   delete batch1;
34   delete batch2;
35
36   return 0;
37 }
```