Question 1 **[25 Marks]**

a. Explain the concept of composition in OOP and describe a real world example that can be applied with this concept. (4 marks)

b. Aggregration is another concept in OOP. When it comes to the implementation for this concept, it is not suitable to use objects, but instead using pointers is more appropriate. Explain why. (4 marks)

c. Consider the class diagram in **Figure 1** which shows the relationship between classes in terms of a composition or an aggregation.

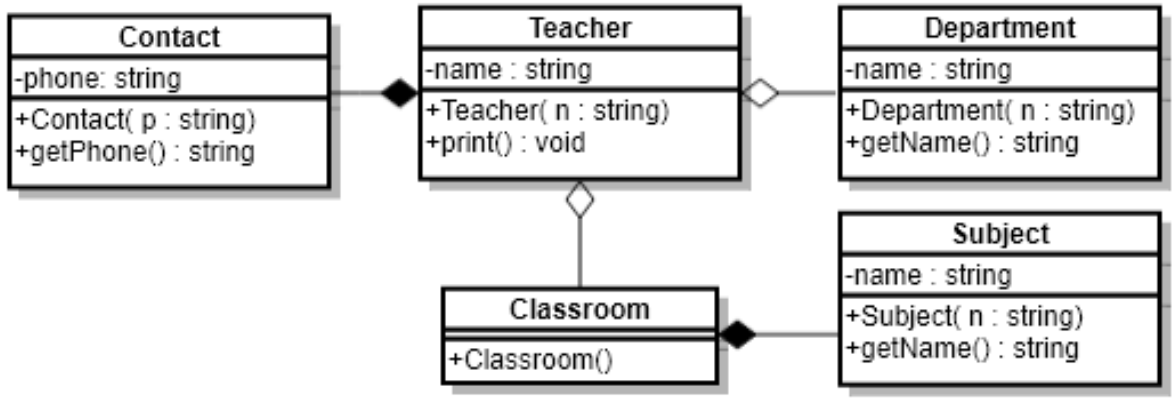

Figure 1

Given **Program 1** below which is meant to implement the class diagram in **Figure 1**. Complete the program with appropriate C++ code for segments **(i)** to **(v)** below.

```
1  //Program 1
2
3
4  #include<iostream>
5  using namespace std;
6
7
8
9  class Contact{
10       private:
11            string phone;
12
13       public:
14            Contact(string p=""){phone = p;}
15            string getPhone() const {return phone;}
16 };
17
18
```

```
19  class Department{
20        private:
21              string name;
22
23        public:
24              Department(string n=""){name=n;}
25              string getName() const {return name;}
26  };
27
28  class Subject{
29        private:
30             string name;
31
32        public:
33              Subject(string n=""){name=n;}
34              string getName() const {return name;}
35  };
36
37  class Classroom{
38        public:
39
40              _______//(i)_____________________________   // (2 marks)
41
42
43              Classroom(){}
44  };
45  class Teacher{
46        private:
47              string name;
48
49              _______//(ii)_____________________________ // (7 marks)
50
51              ___________________________________________
52
53              ___________________________________________
54        public:
55              Teacher(string n=""){name=n;}
56
57              void print() const{
58
59                    string dept = _______//(iii)_____________  // (2.5 marks)
60
61                    string phone= _______//(iv)_____________  // (2.5 marks)
62
63                    string subj = _______//(v)_____________  // (3 marks)
64
65                    cout << "Teacher's name : " << name  << endl;
66                    cout << "Department     : " << dept  << endl;
67                    cout << "Phone number   : " << phone << endl;
68                    cout << "Subject taught : " << subj  << endl;
69              }
70  };


73  int main()
74  {
75        return 0;
    }
```

Question 2 **[35 Marks]**

Consider **Program 2** below and answer questions **(a)** to **(c)**.

```
1  //Program 2
2
3  #include <iostream>
4  using namespace std;
5
6  class Event{
7       public:
8            string venue;
9            Event(){ cout << "Event" <<endl;}
10 };
11
12 class Convocation : public Event{
13      public:
14           int year;
15           Convocation()
16           { cout << "Convo" <<endl;}
17 };
18
19 class Test{
20      public:
21           Test(){ cout << "Test" <<endl;}
22 };
23
24 class Exam: public Test, public Event{
25      protected:
26           int score;
27      public:
28           Exam(){ cout << "Exam" <<endl; }
29 };
30
31 int main()
32 {
33
34    return 0 ;
35 }
```

a. Draw the UML class diagram showing the relationship between the classes in **Program 2**. Notes: as for each class, write only the class name (i.e., without attributes and methods).

(14 marks)

b. What if the following code is added in the main function of the program. Trace the screen ouput with the correct order.

i. `Test t;` // (2 marks)

ii. `Convocation c;` // (4 marks)

iii. `Exam xm;` // (6 marks)

[Notes: 2 marks for each line of output]

c. With reference to the object **xm** in **b(iii)** above, determine the member data (attributes) owned by this object. Also, determine whether the object can access the attributes. Write **Y** in cells **(i)** to **(iii)**, if the object **xm** owns or has a copy of the attribute, and write **N** if otherwise. Similarly, write **Y** in cells **(iv)** to **(vi)** if the object can access the attribute and **N** if cannot.

	venue	**year**	**score**
xm owns	**(i)**	**(ii)**	**(iii)**
xm can access	**(iv)**	**(v)**	**(vi)**

(9 marks) *[Notes: 1.5 marks for each cell]*

Question 3 **[22 Marks]**

Given two classes named `Fruit` and `Banana` and some objects of these classes as shown in in **Program 3** below.

```
 1  //Program 3
 2
 3
 4  #include<iostream>
 5  using namespace std;
 6
 7  class Fruit{
 8
 9     public:
10
11         Fruit(){}
12
13         string malayName(){ return "Buah-buahan";}
14
15         virtual string englishName(){ return "Fruit";}
16
17         void showBenefit(){
18             cout << englishName()  << endl
19                  << "Good for diet" << endl << endl;
20         }
21
22
23         virtual ~Fruit(){
24             cout << "Remove Fruit" << endl;
25         }
26
27
28  }; //End of class Fruit
29
```

```
30 class Banana: public Fruit{
31       public:
32             Banana(){}
33
34             string malayName(){ return "Pisang";}
35             string englishName(){ return "Banana";}
36
37             void showBenefit(){
38                   Fruit::showBenefit();
39
40                   cout << malayName() << endl
41                        <<"High vitamins" << endl;
42             }
43
44             ~Banana(){
45               cout << "Remove Banana" << endl;
46             }
47 }; //End of class Banana
48
49 int main(){
50       Fruit f1;
51       Fruit  *f2 = new Banana();
52       Banana *b  = new Banana();
53
54       // Code for the question will be inserted within these lines.
55
56       return 0;
57 } //End of main()
```

What if the following C++ code are added in the main function at line **54**. Trace the screen ouput with the correct order. Note that, each question below is independent from one to other.

a.
```
cout <<"English: " << f1.englishName() << endl;
cout <<"Malay: "   << f1.malayName()   << endl;
```
// (3 marks)

b.
```
cout <<"English: " << f2->englishName() << endl;
cout <<"Malay: "   << f2->malayName()   << endl;
```
// (3 marks)

c.
```
cout <<"English: "  << b->englishName() << endl;
cout <<"Malay: "    << b->malayName()   << endl;
```
// (3 marks)

d.
```
f2->showBenefit();
```
// (3 marks)

e.
```
b->showBenefit();
```
// (6 marks)

f.
```
delete f2;
```
// (4 marks)

[Notes: 1.5 or 2 marks for each line of output]

Question 4 **[18 Marks]**

Consider **Program 4** below and aswer questions **(a)** to **(g)**.

```
 1  // Program 4
 2
 3
 4  #include<iostream>
 5  using namespace std;
 6
 7  class Operation{
 8        public:
 9              class Infinity{};
10
11              Operation(){}
12
13              int divide(int m, int n){
14                    if (n==0) throw Infinity();
15                    return m / n;
16              } // End of divide()
17
18              void printSequence(int start,int stop){
19                    if (start>stop) throw "Invalid Sequence";
20
21                    int num = start;
22                    while (true){
23                          if (num >= stop) throw num;
24                          cout << "Number: " << num << endl;
25                          num = num + 2;
26                    }
27              } // End of printSequence()
28  }; //End of class Operation
29
30
31
32  int main()
33  {     Operation op;
34        char ch;
35        int n1, n2;
36        try{
37             cout << "Enter a character and "
38                 << "two non-negative numbers => ";
39              cin >> ch >> n1 >> n2;
40
41              if (ch!='D' && ch !='S') throw "Invalid Operation";
42
43              if (n1<0) throw n1;
44              if (n2<0) throw n2;
45
46              if (ch=='D')
47                cout << "Division: " << op.divide(n1, n2) << endl;
48              else
49                  op.printSequence(n1,n2);
50          cout << "Numbers are " << n1 << " and " << n2<< endl;
51        }   //End of try
52        catch (const char *e){
53              cout << e << endl;
54        }
```

```
55          catch (int e){
56                cout << "Numerical Error:  " << e << endl;
57          }
58          catch (...){
59                cout << "Unknown Error" << endl;
60          }
61
61          return 0;
62
63    }   //End of main function
63
```

What would the program print onto the screen if the user enters the following input (i.e., for variables `ch`, `n1`, and `n2`, respectively). Note that, each question **(a)** to **(g)** is an independent run.

a. **D 8 2** (3 marks)

b. **X 3 1** (1.5 marks)

c. **D 5 0** (1.5 marks)

d. **S 1 9** (7.5 marks)

e. **S 7 5** (1.5 marks)

f. **D 4 -2** (1.5 marks)

g. **S -4 1** (1.5 marks)

[Notes: 1.5 marks for each line of output]