INHERITANCE

Question 2 **[30 Marks]**

(i) Inheritance provides a new way to create a new class from an existing class. Give two (2) **advantages** implementing inheritance in programming. (3 marks)

Given Program 2a below and answer question ii) to iv)

```
 1 //Program 2a
 2 #include <iostream>
 3 using namespace std;
 4
 5 class Exam {
 6  private: int year;
 7  public:
 8   Exam()  { cout << "Great"<<endl; }
 9   ~Exam()  { cout << "Positive"<<endl; }
10    void setTheory(double t) {theory = t;}
11    void setPractical(double p){practical = p;}
12    void display(int y){cout<<"This is Exam : "<<y<<endl;}
13    protected: double theory; double practical;
14 }; //Exam
15
16 class Final: public Exam {
17  private: char Ef;
18  public:
19   Final () { cout << "Pray"<<endl; }
20   ~Final(){cout<<"Success"<<endl;}
21   double getFinal(){return (theory + practical);}
22   void myCode(char f)
23    {
24      Ef = f;
25      cout<<"Exam code:"<<Ef<<endl;
26    }
27 }; //Final
28
29 class License: protected Exam {
30  private:
31   string message="Drive";
32  public:
33   void print(string m){
34    cout<<message;
35    cout<<m;}
36  protected :
37   int date, month, year;
38 };//License
39
```

```
40
41
42
43  int main(void) {
44   Exam Eyear;
45   Final F18;
46   Eyear.display(2018);
47   F18.setTheory(40.5);
48   F18.setPractical(25.5);
49   cout<<"Sum: "<<F18.getFinal()<<endl;
50
51   F18.myCode('F');
52   return 0 ;
53  }//main
```

(ii) Draw a UML class diagram to show the relationship between the two classes (`Exam` and `Final`) (3 marks)

(iii) Trace the ouput of Program 2a. Write a <u>complete answers with correct sequence</u> by full fill nine (9) space below: (18 marks)

_____ (1.5 marks)

_____(1.5 marks)

_____(1.5 marks)

_____(2.5 marks)

_____(2.5 marks)

_____(2.5 marks)

_____(2 marks)

_____(2 marks)

_____(2 marks)

iv) Based on Program 2a, write the correct answer to shows the combinations of `Exam` class acess with inheritance access type wether : Not Inherited, protected , public or private (6 marks)

	`Final`	**`License`**
`Exam`	public (access specifier)	protected (access specifier)
private		
protected		
public		