**BACS 387 – Quiz 2 [Fall 2020] [011/20/20]**

This quiz is worth 30 points. You need to create a list of classes that best represent the data below, you must demonstrate one instance for each of abstraction (3 points), encapsulation (10 points) and inheritance (17 points). You must also demonstrate the use of Interfaces as part of inheritance (7 points).

Review the following user-based menu in a console C# application:

1. Enter Students (id, first name, last name)
2. Enter Student grade (e.g, 0-100; ex: 92.5)
   1. Show instructor a list of students
   2. Then instructor selects student by ID to enter grade
3. Remove Students
   1. Show instructor a list of ID + student first name last name
   2. Then instructor may input student id to remove
4. Grade Analytics
   1. Display student grade average
   2. Display Minimum grade + Student Name
   3. Display Maximum grade + Student Name
   4. Display % of A's (90-100)
   5. Display % of B's (80-89)
   6. Display % of C's (70-79)
   7. Display % of D's (60-69)
   8. Display % of F's (0-59)
5. Quit

For this quiz, you must come up with the necessary classes to get this done. You do not have to create methods to demonstrate any calculations, such as those in “Grade Analytics” or “Removal of students”.

You do not need to create any code to demonstrate the usage of your classes.

You are only required to create classes.

Class Student{

Private string ID, FName, LName;

Public student(string id, string Fname, string Lname){

FName = Fname;

Lname = Lname;

ID = id;

}

Public string getID(){return ID;}

}

Class StudentGrade : Student{

Private string ID, FName, LName;

Public int Grade = 0;

Public StudentGrade(string id, string Fname, string Lname, int grade) : base(id, Fname, Lname){

}

Public int getGrade(){ return Grade; }

}

Public interface ListSudents{

Public string listST();

}

Class EnterGrade : ListStudent {

Private int STGrade;

Public List<StudentGrade> Students = new List<StudentGrade>();

Public EnterGrade(int stgrade){

STGrade = stgrade;

}

Public string listST(){

List = “”;

foreach (StudentGrade st in Students){

List = List+ st.getID+ st.getName() ;

}

Return List;

}

Public void setGrade(int id, int grade){

listST();

foreach (StudentGrade st in Students){

If (ST.getid() = id ){

St.grade() = grade;}}

}

}

}

Class RemoveSudent: listST {

Public List<StudentGrade> Students = new List<StudentGrade>();

Public RemoveStudent(List<StudentGrade> students ){

Students = students;

}

Public string listST(){

List = “”;

foreach (StudentGrade st in Students){

List = List+ st.getID+ st.getName() ;

}

Return List;

}

Public void remove(int id){

listST();

foreach (StudentGrade st in Students){

If (ST.getid() = id ){

Students.remove(st);}}

}

Public class GradeAnilitics : listST{

Public List<StudentGrade> Students = new List<StudentGrade>();

Public GradeAnalitics( List<StudentGrade> students){

Students = Students;

}

Public string listST(){

List = “”;

foreach (StudentGrade st in Students){

List = List+ st.getID+ st.getName() ;

}

Return List;

}

}