CS/INFO 1182
Test 2 Practice

Read the instructions carefully and answer each question as completely as you can.

Try doing this test as completely as you can; then you can type the code in to Visual Studio to see if what you expect to happen did.

For the test you will be allowed
One – 3x5 index card

I recommend that this note card is hand written, but it is not required to be so.

CS 1182 Test 2 Practice material

Use the following code to answer the question that follow.

```
namespace TestPracticeCode {
    class Marker<T> {
        public T _Value;
        public T Value { get { return _Value; } set { _Value = value; } }
    }
    interface Named<T> {
        String FullName { get; }
    }
    class Person:Named<Person> {
        protected String _FirstName = "";
        protected String _LastName = "";
        protected int _Age = 0;
        public virtual int Age { get { return _Age; } set { _Age = value; } }
        public String FirstName {
            get { return _FirstName; }
            set { _FirstName = value.Trim(); }
        }
        public String LastName {
            get { return _LastName; }
            set { _LastName = value.Trim(); }
        }
        public String FullName { get { return FirstName + " " + LastName; } }
        public override string ToString() {
            return String.Format("FullName: {0}", FullName);
        }
    }

    class Instructor : Person {
        public Instructor() { _Mark = new Marker<int>(); }
        public List<Student> _Students;
        public List<Student> Students { get { return _Students; }set { _Students = value; } }
        public Marker<int> _Mark { get; set; }
        public Marker<int> Mark { get { return _Mark; } set { _Mark = value; } }
    }

    class Student : Person,IComparable<Student> {
        private String _FavoriteColor;
        private int _ID = 0;
        public Course Course { get; set; }
        private Marker<String> _Mark;
        private StudentType _Type;
        public enum StudentType { Foreign, Domestic, Unknown = -1 }

        public Student() {
            initializeValues("Joe", "Nobody", 0, "Black", int.MaxValue);
        }
        public Student(String newFirstName, String newLastName, int newAge,
            string newColor, int newID) {
            initializeValues(newFirstName, newLastName, newAge, newColor, newID);
        }
        private void initializeValues(String newFirstName,
            String newLastName, int newAge, string newColor,
            int newID) { _Mark = new Marker<string>();
            _Age = newAge; FirstName = newFirstName;
            LastName = newLastName; _FavoriteColor = newColor; _ID = newID;
            _Type = StudentType.Unknown;
        }
```

```
        public String Color {
            get { return _FavoriteColor; }
            set { _FavoriteColor = value; }
        }
        public int ID { get { return _ID; } set { _ID = value > 0 ? value : 0; } }
        public override int  Age { get { return _Age; } set { base.Age = value + 10; } }
        public Marker<String> Mark { get { return _Mark; } set { _Mark = value; } }
        public StudentType Type { get { return _Type; } set { _Type = value; } }
        public override string ToString() {
            return String.Format("Name: {0} Age: {1}" +
                " Color: {2}", FullName, Age, Color);
        }
        public int CompareTo(Student stud) {
            return this.FullName.CompareTo(stud.FullName);
        }
    }
    class Course {
        public Course() {
            Students = new List<Student>();
            People = new List<Person>();
        }

        public int ID;
        public String Name;
        public String IndexNumber;
        public Instructor Professor;
        public List<Student> Students;
        public List<Person> People;
        public String StudentNames() {
            String studNames = "";
            foreach (Student stud in Students) {
                studNames += stud + "\r\n";
            }
            return studNames;
        }
        public String PeopleNames() {
            String pplNames = "";
            Student inst = (Student)People.Last();
            foreach (Person ppl in People) {
                pplNames += ppl + "\r\n";
            }
            return pplNames;
        }
    }
}
```

1. Explain in your own words what is composition?

2. Give an example from the code of composition.

3. What does the following code display?

```
Person pers = new Person();
pers.FirstName = "Bob";
pers.LastName = "Barker";
tbOutput.Text += pers;
```

4. What does the following code display?

```
Student stu = new Student();
stu.Mark.Value = "AF";
tbOutput.Text += stu;
```

5. What does the following code display?

```
Student stud = new Student("Ali", "London", 14, "Purple", 1111);
stud.Age = 16;
if (stud.Type == Student.StudentType.Domestic)
    tbOutput.Text += String.Format("{0} {1} {2}",
        stud.FirstName, stud.LastName, stud.Age);
else if (stud.Type == Student.StudentType.Foreign)
    tbOutput.Text += String.Format("{2} {1} {0}",
        stud.FirstName, stud.LastName, stud.Age);
else
    tbOutput.Text += String.Format("{1} {0} {2}",
        stud.FirstName, stud.LastName, stud.Age);
```

Use this code for the questions 6-11:

```
Student s1 = new Student("Jack", "Sparrow", 100, "Black", 1);
Student s2 = new Student("Will", "Turner", 21, "Brown", 2);
Student s3 = new Student("Elizabeth", "Swann", 18, "Pink", 3);
Student s4 = new Student("Bill", "Turner", 56, "Blue", 4);
Student s5 = new Student("Hector", "Barbossa", 120, "Red", 5);

Course pirating101 = new Course();
pirating101.Name = "How to be a Pirate.";
pirating101.Professor = new Instructor();
pirating101.Professor.FirstName = "Mary";
pirating101.Professor.LastName = "Taylor";
pirating101.Professor.Mark.Value = 100;

s1.Mark.Value = "100";
s2.Mark.Value = "10";
s3.Mark.Value = "30";
s4.Mark.Value = "450";
s5.Mark.Value = "200";

pirating101.Students.Add(s1);
pirating101.Students.Add(s4);
pirating101.Students.Add(s2);
pirating101.Students.Add(s3);
pirating101.Students.Add(s5);
s1.Course = s2.Course = s3.Course = s4.Course = s5.Course = pirating101;
```

6. What does the following code display?

```
tbOutput.Text += pirating101.StudentNames();
```

7. What does the following code display?

```
tbOutput.Text += pirating101.Students[4].Course.Name;
```

8. What does the following code display?

```
pirating101.Students[4].Course.Name = "Swashbuckling 101";
tbOutput.Text += pirating101.Students[3].Course.Name;
```

9. What does the following code display?

```
tbOutput.Text += pirating101.Students[1].Course.Professor.Mark.Value
    + pirating101.Students[2].Mark.Value;
```

10. What does the following code display?

```
pirating101.Students.Sort();
       tbOutput.Text += pirating101.StudentNames();
```

11. Write a C# method that accepts the course object pirating101 and displays to the TextBlock the name, and mark of each student and their number in the list with the name and mark of the professor which should look like the following:

    1. Turner, Bill -450- : Prof. Taylor "100"
    2. Swann, Elizabeth -30- : Prof. Taylor "100"
    3. Barbossa, Hector -200- : Prof. Taylor "100"
    4. Sparrow, Jack -100- : Prof. Taylor "100"
    5. Turner, Will -10- : Prof. Taylor "100"