CS1102S Data Structures and Algorithms

Assignment CC01: Object-oriented Programming

1 Classification

Define a class Circle that allows for creating a circle with a given radius:

```
Circle myCircle = new Circle(7.5);
```

Circle objects should support the following operations:

- retrieve the radius of the circle
- set the radius of the circle to a new value
- retrieve the circumference of the circle

Solve this task using two different approaches:

1. Use a public field "radius" for the retrieval and setting of the radius, and use a static function to retrieve the circumference.
2. Use function "getRadius" for the retrieval and setting of the radius (where the radius is kept in a private field), and use a non-static function to retrieve the circumference.

2 Aggregation

Define a class Person that allows for creating an person object with a given name, year of birth, and address:

```
Person myPerson = new Person("Tan Ka Boon",1975,
                             "107 Pasir Ris Avenue 6; #03-01");
```

Person objects should support the following operations:

```
myPerson.move("10 Clementi Avenue 5; #07-05");    // return nothing
String s = myPerson.getAddress();
int age = myPerson.getAge();   // assume it's midnight 1/1/2009
```

3 Inheritance

Define a class Student that supports the above operations on persons, but in addition provides the following operations:

```
Student myStudent = new Student("Tan Ka Boon",1975,
                           "107 Pasir Ris Avenue 6; #03-01");
// register a student with a given matric number
myStudent.register("U3455697X");
// return the matric number
String s = myStudent.getMatriculationNumber();
```

4 Java Specifics

1. Verify that for method overriding, it is not enough that the argument types of the overriding method are sub-types of the argument types of the overridden method. They must indeed be *exactly* the same.

2. Verify that method dispatching in Java uses the class of the actual object, not the declared type of the object.

3. Verify that method dispatching in Java uses the declared types of the arguments, and not the actual classes of the arguments.

4. Verify that array types of Java are co-variant, and produce a runtime error as a result of this co-variance.

5. (Challenge question) The newest version of Java provides for return type co-variance. Find out in the web what return type co-variance is and give an example.