

CS 535 Object-Oriented Programming & Design
Fall Semester, 2001
Doc 6 Assignment 1 Comments
Contents

Issues 4
 Calculating Distance between Points 4
 includes: 4
 Why origin:radius: and radius:origin:?. 6
 Once and Only Once..... 6
 Names..... 7
 Comments..... 8

One Possible Solution

```
Smalltalk.CS535 defineClass: #SimpleCircle
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'origin radius '
  classInstanceVariableNames: "
  imports: "
  category: 'Course-Examples'
```

```
CS535.SimpleCircle class methodsFor: 'instance creation'
```

```
origin: aPoint radius: aNumber
  ^super new
    setOrigin: aPoint
    setRadius: aNumber
```

```
radius: aNumber origin: aPoint
  ^self
    origin: aPoint
    radius: aNumber
```

CS535.SimpleCircle methodsFor: 'accessing'

area

\wedge radius * radius * Float pi

includes: aPoint

\wedge (origin dist: aPoint) <= radius

origin

\wedge origin

radius

\wedge radius

CS535.SimpleCircle methodsFor: 'initialize'

setOrigin: aPoint setRadius: aNumber

origin := aPoint.

radius := aNumber

Issues

Keep in mind this was the first assignment in a language you are starting to learn. Some of the points covered below I did not expect people to know. However, since some solutions turned in raised these issues I discuss them here.

Calculating Distance between Points

Some people used the method "dist:" in the Point class that computes distance between two points others did not. I suspect that those that did not use it did not find it.

In Smalltalk all work is done by sending a message to an object. Since we are dealing with Point objects the place to look is the Point class. Finding Point class and searching its methods requires knowing the how to use VisualWorks environment. Often a Smalltalk programmer will spend more time looking for code to use than writing new code.

includes:

Student solution A.

```
includes: aPoint
```

```
(origin dist: aPoint) <= radius
```

```
  ifTrue: [Transcript show: 'the point is in the circle']
```

```
  ifFalse: [Transcript show: 'the point is outside the circle']
```

This solution sends the result to the Transcript instead of returning the result. This is not what was asked in the assignment. Such code can not be used anywhere else. Say for example in a program that determined if a coordinate is within a given distance of a building. When you compute a value in a method return the value. If the program wants to print the value on the Transcript (or to a file, display in a GUI widget, etc) it can do that.

Student solution B.

```
includes: aPoint
  (origin dist: aPoint) <= radius
    ifTrue: [^True]
    ifFalse: [^False]
```

Smalltalk uses true and false (starting with lower case characters) for boolean values. These are unique instances of the True and False classes.

Student solution C.

```
includes: aPoint
  (origin dist: aPoint) <= radius
    ifTrue: [^true]
    ifFalse: [^false]
```

Here the correct values are being returned, but the code is longer than needed. The following is shorter and takes less time to understand.

```
includes: aPoint
  ^(origin dist: aPoint) <= radius
```

Why origin:radius: and radius:origin:?

Both methods were given just to show that one could have multiple methods to create objects. Users of the class do not have to remember which order is correct. However for three or more arguments providing all possible permutations becomes too much.

Once and Only Once

origin: aPoint radius: aNumber

^super new

setOrigin: aPoint

setRadius: aNumber

radius: aNumber origin: aPoint

^self

origin: aPoint

radius: aNumber

Do things only once. Since origin:radius: creates and initializes a SimpleCircle object, radius:origin: calls

Names

setOrigin: t1 radius: t2

origin := t1.

radius := t2

includes: aPoint

| t1 t2 t3 t4 |

t1 := origin x.

t2 := origin y.

etc.

Names should help convey meaning to the programmer

In Smalltalk

Use type information for arguments

Use role of the variable for local variables

Comments

origin

"return the origin"

^origin

Comments are to help a reader understand the code

Does the comment in the code above help?

Ralph Johnson at UIUC subtracts points from student's work when they comment methods that just set and get values.