CSE 331 SECTION 4: ABSTRACT DATA TYPES

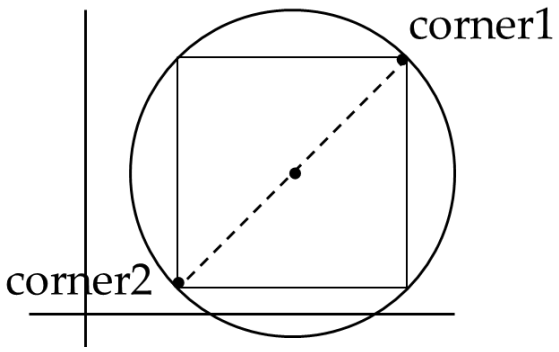
Notes and Definitions

Abstract Representation: ADTs	Concrete Representation: Data Structures
1. **Abstract State:** What does the state of the data *represent*? What do the **fields** represent?	1. **Concrete State:** What *is* the state of the data? What are the **fields**?
2. **Abstract Operations:** *What* operations can you do with the data? What **methods** are present, and what do they do?	2. **Concrete Operations:** *How* do you implement those operations to do that? How do you implement those **methods**?
•How the **client** views the data: ◦ Independent of underlying code	•How the **implementer** views the data: ◦ The actual underlying code

- Abstraction Function: Maps Concrete State to Abstract State
- Representation Invariant: Maps Concrete State to Boolean
  - TRUE iff Abstraction Function holds

Problems

1. Fill in the abstraction function and representation invariant for this implementation of Circle. Suppose our concrete representation in this case is two points directly across from each other, representing the endpoints of a diameter of the circle.




```
public class Circle3 {
     private Point corner1, corner2;

     // Abstraction function:

     // AF(this) = a circle c with center (x,y) and

     // radius r such that

     //      (x,y) = _____________________________________________

     //      r = _____________________________________________

     // Rep invariant:

     // _____________________________________________________________

     // _____________________________________________________________

}
```

2. Given the following ADT, NonNullStringList, find two concrete representations for it. NonNullStringList is a list of `string` such that there are no `null` values in the list. Note your implementations must have

some way to implement the three abstract operations provided (`add`, `remove`, `get`). Write out the abstraction function and representation invariant for both.
Hint: Recall the two implementations of List.

Concrete Representation 1:

```
public class NonNullStringList  {
     // Abstraction function:









     // Rep invariant:









     // Fields:




     public void add(String s) { ... }
     public boolean remove(String s) { ... }
     public String get(int i) { ... }
}
```

Concrete Representation 2:

```
public class NonNullStringList  {
     // Abstraction function:









     // Rep invariant:









     // Fields




     public void add(String s) { ... }
     public boolean remove(String s) { ... }
     public String get(int i) { ... }
}
```