SE1021 Exam 1 Name:

Use only your pen/pencil/eraser (for example, no note-sheet). Review all questions before you get started. The exam is printed double-sided. Show all work. You do not need to comment your code.

Problems 1-4: [20 pts.]

1. (4 pts.) What does the following code-snippet print when it is run?

```
int x = 0;
System.out.println(++x);
System.out.println(x++);
System.out.println(++x);
```

2. (4 pts.) What does the following code-snippet print when it is run?

```
char c1 = ‘A’;
char c2 = (char)(c1 + 1);
System.out.println("c2: "+c2);
```

3. (6 pts.) Re-write the following code so that it does not crash if `tooter` is null. If `tooter` is null, the code should simply skip the if block.

```
if(tooter.hasNext()) {
  T00t toot = tooter.next();
}
```

4. (6 pts.) Suppose that a House class extends a Building class, but the Building classes’ only constructor is:

```
public Building(int squareFeet) {
    this.squareFeet = squareFeet;
}
```

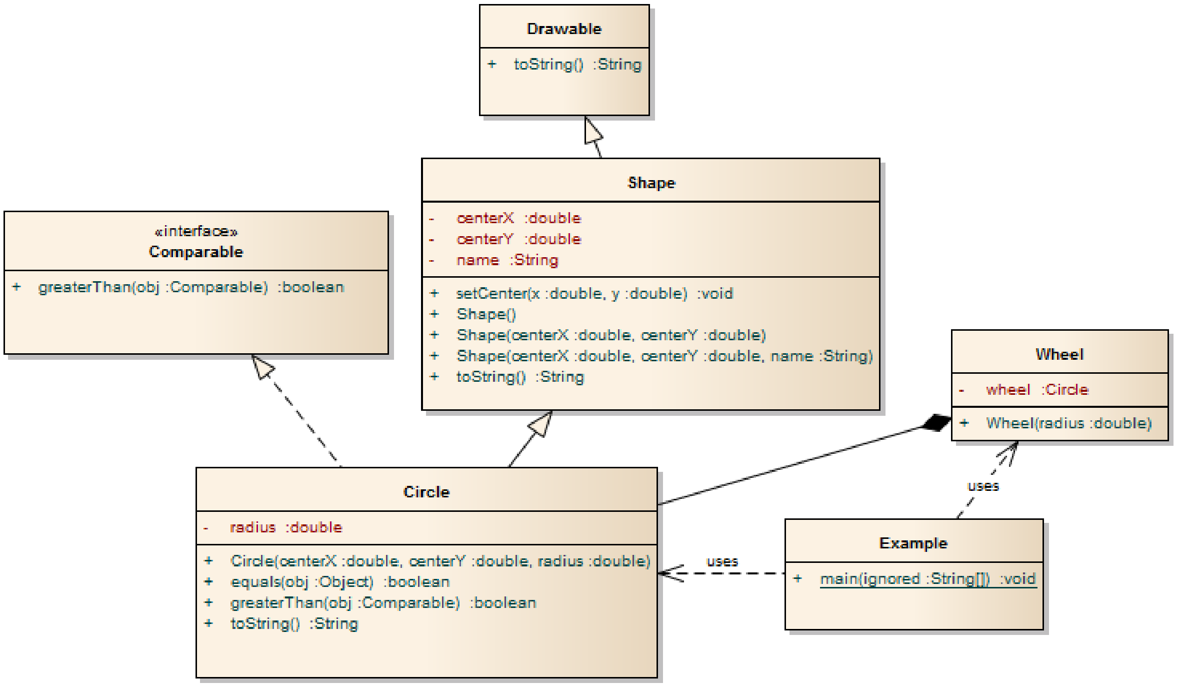
Complete the constructor for the House class:

```
public class House extends Building {
    private int numBaths;
    public House (int numBaths, int squareFeet) {



    }
}
```

Questions 5-8 [8pts.] For these questions, refer to this UML diagram, and select one answer.


Drawable
+ toString() :String
Shape
- centerX :double
- centerY :double
- name :String
+ setCenter(x :double, y :double) :void
+ Shape()
+ Shape(centerX :double, centerY :double)
+ Shape(centerX :double, centerY :double, name :String)
+ toString() :String
«interface»
Comparable
+ greaterThan(obj :Comparable) :boolean
Wheel
- wheel :Circle
+ Wheel(radius :double)
uses
Circle
- radius :double
+ Circle(centerX :double, centerY :double, radius :double)
+ equals(obj :Object) :boolean
+ greaterThan(obj :Comparable) :boolean
+ toString() :String
uses
Example
+ main(ignored :String[]) :void

5. (2 pt.) In the relationship between Circle and Shape, **Shape** is the
   a. base class
   b. derived class
6. (2 pt.) The relationship between Circle and Shape is
   a. Inheritance
   b. Aggregation
   c. Composition
   d. Implementation
7. (2 pt.) The relationship between Comparable and Circle is
   a. Inheritance
   b. Aggregation
   c. Composition
   d. Implementation
8. (2 pt.) The relationship between Circle and Wheel is
   a. Inheritance
   b. Aggregation
   c. Composition
   d. Implementation

Problems 9 & 10: [30 pts.]

9. (18 pts.) Typecasting errors are the only errors in this problem. (You may assume the appropriate includes have been done.) Write the types of the left and right hand side of the assignment **that is in bold**. For non-primitive types, write "ref" for a reference. Some of the classes are from the UML diagram on the previous page. The two rows are filled in as an example.

	Code Snippet	**Type of left-hand-side**	**Type of right-hand-side**	**Will it compile?**

(ex)	float f = 4.0f;
	double d = f;
		double	float	yes

a	**float f = (long)1e9;**			

b	String str = "asdf";
	char c = str;			

(ex)	**String str = new Object();**	String ref	Object ref	no
c	**Object o = new String();**			
d	**Comparable c = new Circle();**			

e	Shape s = new Circle(0,0,1);
	Comparable c = s;			

f	Scanner in = new Scanner(System.in);
	double x = in.nextInt();			

10. (12 pts.) This problem is the same as problem 5, except now you should the type expected by each method for its parameter, and the type provided by the function calling it.
For all examples, assume Circle c1 = new Circle(0,0,1);

	Code Snippet	**parameter expected**	**type given**	**Will it compile?**

(ex)	Object c2 = (Object) new Circle(0,0,1);
	c1.equals(c2)
		Object ref	Object ref	yes

a	Shape c2 = new Circle(0,0,1);
	c1.greaterThan(c2)			

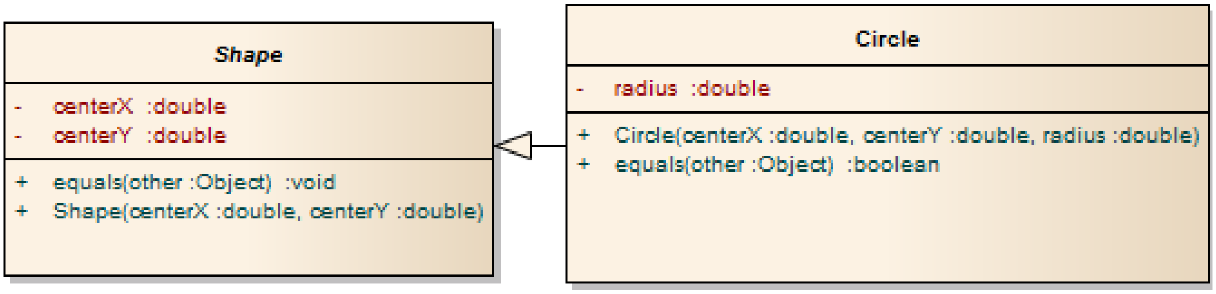
b	Wheel c1 = new **Wheel(1L);**			

c	Wheel c2 = new Wheel(1.0);
	c1.greaterThan(c2)			

d	Scanner in = new Scanner(System.in);
	double x = in.nextInt();			

11. (20 pts.) Assume that the Shape class is implemented as discussed in class. Implement the Circle class without modifying anything in the shape class. The `equals()` method should check that `centerX` and `centerY` are equal as well as radius. You may assume that the “equals” method of Shape returns “true” if `centerX` and `centerY` are both equal.

If you are not sure where to get started: Write the circle class with empty methods with lots of room in them. Then write what should be in the methods.


Shape
- centerX :double
- centerY :double
+ equals(other :Object) :void
+ Shape(centerX :double, centerY :double)
Circle
- radius :double
+ Circle(centerX :double, centerY :double, radius :double)
+ equals(other :Object) :boolean

Circle.java (write your code below this line)