Basic Object Orientation

CSC02A2

UNIVERSITY OF JOHANNESBURG

Outline

Outline

1. Objects and Classes
   - OO Metaphor
   - Principals of OO Design
   - Java Classes and Objects
   - Constructors and member access
   - Reference types and Garbage Collection
   - Modifiers
   - Scope in a class
   - The `this` reference
   - Packages
   - Immutable Objects
   - UML Class Diagrams

Objects and Classes

Object Orientation Metaphor

Problems are modeled as sets of interacting objects. Objects are defined as having **state** and **behaviour**.

Object state The combined values of the attributes which the object is made up of.

Object behaviour The actions which are available to the object which operate on the object's state.

- A **class** is a *template* or *blueprint* for an object.
- An object is an *instance* of a class.

Principals of Object Orientated Design

Divide and Conquer Decompose overall problem into a set of interfacing objects.

Encapsulation Assign attributes and actions to an object. Object is a self contained module with a clear responsibility.

Interface Clarify interactions between objects.

Information Hiding Design and implementation specific details are hidden from users of the object.

Generality Design classes for a particular kind of task.

Extensibility Design with future expansion in mind.

Abstraction Focus on important and relevant features of an object.

Java Classes and Objects

Java classes define the attributes and actions which an instance of the class can have.

- Attributes are known as *instance variables*
- Actions are known as *instance methods*.

Java classes can also contain *class variables* and *class methods* which belong to a class as a whole. Class variables and class methods have the `static` modifier.

Constructors and member access

- Each class has a default no-arguments constructor which is provided by the Java language.
- The name of the constructor is the same name as the class.
- Constructor has no return type specified.

Creating a new instance of a class requires the use of the `new` keyword.

```java
ClassName instance = new ClassName();
```

Instance variables and method called are accessed by the member access operator.

```java
instance.getVariable();
```

Reference types and Garbage Collection

The keyword `null` can be assigned a reference variable which indicates that the reference variable does not point to anything in memory.

Accessing a `null` reference will result in a `NullPointerException`.

Instance variables are initialised to default values but temporary variables are not.

If an instance of an object is no longer referenced in any reference variables then the garbage collector will delete the object to free up memory.

Modifiers

Visibility modifiers within a class:

- `public` - Visible to all
- `private` - Visible within the class only.
- `protected` - Visible within the class and sub-classes.
- not specified - Visible within all class in the same package.

Other modifiers within a class:

- `static` methods - method can be called without an instance
- `static` variables - only one copy of the variable in memory.
- `final` methods - method cannot be overridden (discussed in advanced OO).

Scope of variables in a class

- The scope of a class' instance variables is the entire class regardless of where the variables are declared.
- If a second variable depends on the value of another variables then that variable must be declared first.
- For consistency sake it is considered good practice to declare a class' variables at the beginning of the class.
- Local variables with the same name as a class' fields will overshadow those fields.

The `this` reference

The `this` keyword is a special reference to the current instance of the class.

- Commonly used to access hidden field data caused by overshadowing.
- Used within a constructor to invoke another of the class' constructors with a different number of arguments. This statement must be the first statement in a constructor should it appear.

Packages

- Related classes may be placed into a collection known as a package.
- Packages are usually named after the URL of the company in reverse order.
- `package za.ac.uj.acsse.csc2a;`
- The full name of a class will be `za.ac.uj.acsse.csc2a.ClassName` and the compiled class file will be placed in `za/ac/uj/acsse/csc2a` directory.

Immutable Objects

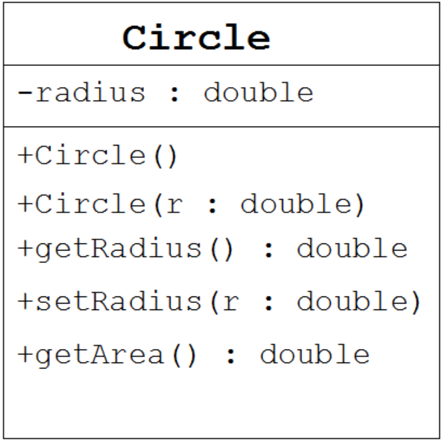
While the state of most objects changes during program execution, occasionally the need for objects whose state remains constant throughout their lifetime.

Strings are immutable (will be discussed in detail in the next lecture).

For a class to be immutable it must meet the following criteria:

- All instance variables are declared private.
- No mutator methods.
- No accessor methods that returns a reference to a mutable data type.

UML Class Diagrams



```
Circle
-radius : double
+Circle()
+Circle(r : double)
+getRadius() : double
+setRadius(r : double)
+getArea() : double
```

```java
public class Circle
{
  private double radius = 1.0;
  public Circle() {}
  public Circle(double r)
  {
    setRadius(r);
  }
  public double getRadius()
  {
    return radius;
  }
  public void setRadius(double r)
  {
    radius = r;
  }
  public double getArea()
  {
    return radius * radius * Math.PI;
  }
}
```