Review for Midterm 1

T. M. Murali

Feb 18, 2004
Objects and Classes

- **Object**: represents “things” from the real world or from some problem domain.
- **Class**: represents all objects of a type.
  - An object is an **instance** of a class.
  - A class can have multiple instances.
- We can **send messages** to objects by **invoking** their **methods**.
Object Methods

- Methods can have 0 or more parameters that pass additional information for the object to use when executing the method.
- Parameters have types (e.g., boolean, int, string).
- A method’s signature consists of its name and the types of its parameters.
- A method can return a value of a particular type.
- A class can have multiple methods with the same name (method overloading)
- All instances of a class have the same attributes or fields.
- Each object has its own set of values, called the state of the object.
Defining Classes

- A class has three parts: name, fields, and methods.
- **Constructor** is a special method that sets up an object correctly when we create it.
- **Scope** of a variable is the portion of the source code from where it can be accessed.
- **Lifetime** of a variable is the period of time it exists.
- **Accessor** methods return information about an object.
- **Mutator** methods change the state of an object.
- **Local variable**: declared and used within a method.
Object Interaction

- **Abstraction** and **Modularisation**
- Classes define types; class name is used as the type of a variable.
- Java has two kinds of types
  - **Primitive types**: int, boolean, float
  - **Classes**: we can create instances of classes
- A class can use other classes.
  - Use `new` to create an instance of a class.
  - Java keyword null means “no object”.
- **Class diagram** shows the classes of an application and relationships between them.
- **Object diagram** shows objects in an application and their relationships at a given time.
Grouping Objects

- Many applications need collections of objects, with number of objects unknown in advance.
- Java provides a rich and powerful API for Collections, part of Java standard class library.
- Use Java documentation to find the methods in a class.
- ArrayList collection
  - Can store unlimited number of objects of any type.
  - Cannot store a primitive type.
  - Use while/for loops to examine each object in the ArrayList.
Sophisticated Behaviour

- Use Iterators to iterate over collections.
- Java array is a fixed-size collection. You can and must specify type of object stored in the array.
- Use Java class libraries.
- Use Java documentation to find the methods in a class.
- Use import statement to import a class in a library into the current scope.

String class

- toLowerCase(), trim() are examples of methods in the String class.
- A String is an immutable object; once a String is created, a method in the String class cannot modify the instance.

- String equality: == versus equals().
Documentation

- **Interface** of a class describes what the class does and how it can be used.
- **Implementation** of a class is the source code that defines a class.
- Elements of documentation
- Writing class documentation (javadoc tool).
Java Library Classes

- Random
- HashMap
- HashSet
- StringTokenizer
- String
Modifiers

- private vs. public variables
- static variables
- final variables
Well-Behaved Objects

- Testing: unit testing.
- Debugging and debuggers
- Test automation
- Writing for maintainability