Computer Science II
Data Structures

Instructor
Sukumar Ghosh
201P Maclean Hall
Office hours: 10:30 AM – 12:00 PM
Mondays and Fridays

Course Webpage
homepage.cs.uiowa.edu/~ghosh/2116.html
Course Syllabus

• Constructs in Java, the language we will use
• Algorithm complexity and Big-O notation
• Arrays, Linked lists
• Solving problems using recursion
• Stacks, queues, lists and trees
• Searching and sorting
• Priority queues, hash tables, binary search trees
• Graphs and basic algorithms on graphs

Teaching Assistants

• Kyle Diedrich
• Adrian Pereira
• Thamer Alsulaiman
• Dhuv Vyas
**CS 2230: Computer Science II: Data Structures**

**About this course**

Main class (AAA) and six sections A01-A06
Discussion sections meet on Thursdays only.
You must go to your own section.

**Textbook**


**Prerequisites**

Computer Science I (CS: 1210 / 22C:016/ ENGR 2730)
(Note: CS 2210: Discrete Structures is a corequisite, if not taken as a prerequisite earlier)
Grading

Eight Home assignments (30%)

Two quizzes (2x5% = 10%)

Two in-class midterms (2x20% = 40%), and

(\textit{Wednesday, Sep 26} and on \textit{Wednesday, Oct 31})

One Final exam (20%)

Limited collaboration is OK, assuming you have first spent some time (about 60 minutes) working on the problem yourself. However, your solution should not be a copy (whole or in part) of a fellow student.

Late Homework Policy

\textit{Quota of two days} for the entire semester
How Java works

Java program

Compiler

Bytecode

Java Virtual Machine

Result
Which IDE will we use?

We will use NetBeans.

You can download it on your machines. They are installed in all lab machines. WE will demonstrate it today in the class.
Object-oriented programming

An Object is a repository of data

Typed data. Always declare before you use.
Primitive types. int, char, boolean, float etc

Class. A template for creating objects

Think of shoppingList as a class. We can define two objects for this class.

Example:

<table>
<thead>
<tr>
<th>sugar</th>
<th>pencil</th>
</tr>
</thead>
<tbody>
<tr>
<td>coffee</td>
<td>paper</td>
</tr>
<tr>
<td>milk</td>
<td>stapler</td>
</tr>
</tbody>
</table>

myList

yourList

Object myList belongs to the Class ShoppingList.
A Class denotes the type of object
y = new Counter();

public class Counter
private int count
public Counter() {}

public

y

reference
Class Human{
    int age;
    String name;
Public void introduce() {
    System.out.println("I’m" + name + "and I’m" + age + " years old")
    }
}

Now, continue as

```java
Human Alice = new Human(); // Create Alice
Alice.age = 21; // Set Alice’s fields
Alice.name = "Alice";
Alice.introduce();
```
# Structure of a Java program

```java
public class MyFirstJavaProgram {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

<table>
<thead>
<tr>
<th>Class</th>
<th>blueprint of an object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class name starts with a capital letter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>instance of a class, created using a constructor</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Instance variables</th>
<th>Unique set of variables for an object</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Actions to manipulate data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method name starts with lower case letters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Program file name</th>
<th>Must exactly match the class name. Saved as filename.java</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>a group of related class definitions</th>
</tr>
</thead>
</table>
What are public, private, protected?

These are Access Control Modifiers.

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Same class</th>
<th>Same package</th>
<th>Subclass</th>
<th>Universe</th>
</tr>
</thead>
<tbody>
<tr>
<td>private</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>protected</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>public</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Organization of a class

1. Constants
2. Instance variables
3. Constructors
4. Methods

```java
public class Counter {
    private int count; // a simple integer instance variable
    public Counter() { } // default constructor (count is 0)
    public Counter(int initial) { count = initial; } // an alternate constructor
    public int getCount() { return count; } // an accessor method
    public void increment() { count++; } // an update method
    public void increment(int delta) { count += delta; } // an update method
    public void reset() { count = 0; } // an update method
}
```
**Primitive types.** Integer, Boolean, character etc

**Reference variables.** A variable whose type is a class is called a reference variable.

c = new Car()

myList = new ShoppingList()
Declare before use

All objects must be declared before they are used.

Consider the following:

```java
Rectangle R;
int w = R.getWidth(); // This is wrong!
```

It will not work since R is just the name of a handle. The object has not been created yet.

```java
Rectangle R;
R = new Rectangle(5,-10,100,200);
int w = R.getWidth(); // This is OK
```