Programming Review

(a teensy tiny little short one)
Assumption

• I assume you can write code!
  • So I won’t do a full review.

• If need a brush up
  – sit in on Intro to Programming.
  – read my online programming notes.
  – read programming text book.
  – get a personal tutor!
  – take advantage of CS department tutors!

• Practice, practice, practice.
Writing code

• **Language**
  - We’ll use **Java** – object oriented, portable.
  - Other languages would work – data structures are language independent!

• **Paradigm**
  - Use object-oriented methodologies

Don’t sweat OO too much. I’ll help.
Writing code: style

• Many brief informative comments!
• Place { and } on their own lines.
  
```java
if(x == 0)
{
   …stuff…
}
```
• **Indent** body of anything that has or could have {…stuff…}
• **Use descriptive variables.**
  
```java
boolean badHairDay (good)
boolean bhd (bad)
```
/** Stores integer data. */
public class DataStore
{
    //private internal storage, not directly accessed
    private int data = 0;

    /** return the stored object – getter method */
    public int getData()
    {
        return data;
    }

    /** store an object – setter method */
    public void setData(int theData)
    {
        data = theData;
    }
}
/** Stores any kind of object. */
public class DataStore {
    //private internal storage, not directly accessed
    private Object data = null;

    /** return the stored object – getter method */
    public Object getData()
    {
        return data;
    }

    /** store an object – setter method */
    public void setData(Object theData)
    {
        data = theData;
    }
}

note: private instance variable

If you have only taken intro programming, then this will look familiar, but advanced. Don’t worry – there will be lots of help and examples.
Bubblesort Example 1

• Let’s practice writing code with Bubblesort
  • Need three primary parts
    1. Put numbers into array.
    2. Scan left to right, and swap if out of order.
    3. Repeat scans until done.
Bubblesort Example 2

• Start with pseudo-code

```c
main
{
    define array = {1,3,2,7,3,9,0}; //part 1
    while still needs sorting //part 3
    {
        compare and sort pairs //part 2
    }
    print answer //extra, but need!
}
```

• Now refine.
Bubblesort Example 3

- Refined pseudo-code
  ```
  main
  {
    array = {1,3,2,7,3,9,0};
    while notDone
    {
      for(each element of array)
      {
        if(next element is smaller)
        {
          swap elements
        }
      }
    }
    print array
  }
  ```

- Next refine even more!
main()
{
    //would be nicer if read these from user
    naArray = {1, 4, 3, 6, 5, 2, 7};

    //loop until no changes happen
    while(bNotDone)
    {
        for(int i=0; i<naArray.length; i++)
        {
            if(naArray[i] > naArray[i+1])
            {

                //you finish the swapping part
                …swap the values…

                //had to swap a value,
                //so not done yet
                bNotDone = true;
            }
        }
    }
}

• What still needs doing?
  – add necessary class stuff
    • in Java, “public class Bubblesort”, etc.
    • in C, “#include <stdio.out>”, etc.

  – declare variables

  – swap the array values
    • put naArray[i] into naArray[i+1] and vice versa

  – output the start and finish values
    • otherwise don’t know what it did!

Watch out!
i+1 might get too big.
Writing Efficient Code

• This class focuses on making *algorithms* more efficient.
  • As a consequence, your code gets more efficient.

• We do not talk about specific compiler and language tricks.
  • Generally, those tricks are small potatoes compared to writing a bad algorithm.