

CS 635 Advanced Object-Oriented Design & Programming
Fall Semester, 2018
Doc 4 Assignment 1 Comments
Sep 11, 2018

Don't Panic

This assignment is 1% of your grade

How many people have trained in one of?

Sports

Musical instruments

Dance

Debate

Chess

Art

Public speaking

Performance art

How important is practice?

Comments

What is the purpose of comments?

```
x = x + 1 // Add one to x
```

Why are we adding one to x?

Duh Comments

Repeating code in English

Commenting the obvious

Comments

```
// Class to represent Priority Queue
class PriorityQueue {
    // fields
    private Student[ ] studentHeap;
    private int heapSize;

    // Constructor to create the priority queue
    public PriorityQueue(int capacity) { ... }

    // get size of the heap
    public int size() {
        return heapSize;
    }
}
```

Comments

```
// get size of the heap  
public int size() {  
    return heapSize;  
}
```

```
/**  
 * return size of the queue  
 */  
public int size() {  
    return heapSize;  
}
```

```
class Student {  
  
    public double priority() {  
        return this.units * 0.7/150 + this.gpa * 0.3/4.0;  
    }  
}
```

What are

0.7

150

0.3

4.0

```
class Student {  
  
    public double priority() {  
        return this.units * UNITS_WEIGHT/MAX_UNITS +  
            this.gpa * GPA_WEIGHT/MAX_GPA;  
    }  
}
```


Comments

What needs commenting in this assignment?

Why did most people leave the first element of array/array list empty?

Struct

Only works this assignment

Hides priority computation

How to find it?

```
class Element {  
    private Student student;  
    private double priority;  
  
    public Element(Student student) {  
        this.student = student;  
        this.priority = student.units() * 0.7 + student.gpa() * 0.3;  
    }  
  
    public Student getStudent() { return student;}  
  
    public double getPriority() { return priority;}
```

```
class Association<K,V> {  
    private K key;  
    private V value;  
  
    public Association(K key, V value) {  
        this.key  
    }  
  
    public key() { return key; }  
  
    public value() { return value; }
```

Independent of student & assignment

Usable in

- hashtable/dictionary
- priority queue
- binary tree

Code Reuse

```
class PriorityQueue {  
    private Student[ ] studentHeap;
```

Where can this be used?

```
class PriorityQueue {  
    private Student[ ] studentHeap;
```

Where can this be used?

```
    private double studentPriority(Student aStudent) {  
        return 0.3 * aStudent.gpa()/4.0 + 0.7 * aStudent.units()/150;  
    }
```

```
class PriorityQueue<E> {  
    private Association<Double, E>[ ] heap;
```

Where can this be used?

```
class PriorityQueue:

    def add_node(self, input):
        if type(input) is not Student:
            raise ValueError("Must be a Student")

        etc
```

How do Python collection classes handle this?

```
class PriorityQueue:

    def addNode(self, input):
        if type(input) is not Student:
            raise ValueError("Must be a Student")

        etc
```

What does Pep 8 say about naming methods?

Responsibility

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    private double studentPriority(Student aStudent) {  
        return 0.3 * aStudent.gpa()/4.0 + 0.7 * aStudent.units()/150;  
    }  
}
```

Why is PriorityQueue computing priority of a student?

studentPriority does not use any field of PriorityQueue

```
class Student {  
    private int units;  
    private double gpa;  
  
    public double priority() {  
        return 0.3 * gpa()/4.0 + 0.7 * units()/150;  
    }  
}
```

Why this priority?

Why only one priority?

Priority depends on the context of the object


```
class PriorityQueue {
    private Student[] studentHeap;

    private int checkGpaRange(double g) {
        int checkgpa = 1;
        if (!(g >= 0.0 && g <= 4.0)) {
            System.out.println("\nInvalid Input");
            checkgpa = 0;
            return checkgpa;
        } else {
            checkgpa = 1;
            return checkgpa;
        }
    }
}
```

Why is PriorityQueue validating student's data?

```
private int checkGpaRange(double g) {
    if (!(g >= 0.0 && g <= 4.0)) {
        return 0;
    } else {
        return 1;
    }
}
```

```
private int checkGpaRange(double gpa) {
    return gpa >= 0.0 && gpa <= 4.0;
}
```

```
class PriorityQueue {  
    private static Student[ ] studentHeap;
```

```
PriorityQueue a = new PriorityQueue();  
a.add(student1);  
a.add(student2);
```

```
PriorityQueue b = new PriorityQueue();  
b.size(); // 2
```

```
class PriorityQueue {  
    static int size = 0;  
    private ArrayList<Student> studentList = new ArrayList<Student>();
```

size will not be consistent with size in studentList

studentList knows its size

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public void askForInput() {  
        promptUserForStudentInfo():  
        Student newStudent = readStudentInfoFromComandLine();  
        add(newStudent);  
        print();  
    }  
}
```

```
class Student {
    public Student(String name, int redID, String email, float GPA, int unitsTaken) {
        Scanner in = new Scanner(System.in);
        this.name = name;
        this.redID = redID;
        this.email = email;

        while (GPA < 0.0f || GPA > 4.0f) {
            System.out.print("Please enter a GPA between 0.0 and 4.0: ");
            GPA = in.nextFloat();
        }
    }
}
```

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public void print() {  
        for (student in list) {  
            System.out.println(student.toString());  
        }  
    }  
}
```

How many Java JDK classes have methods that print to standard out?

Can not test the result

Can not use the print method in
mobile
web
back end
enterprise
GUI app
Embedded

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public String toString() {  
        blah  
        return result;  
    }  
}
```

Can test the result

Can use the toString method in
mobile
web
back end
enterprise
GUI app
Embedded

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public Iterator iterator() {  
        blah  
        return iterator;  
    }  
}
```

Can test the result

Can use anywhere

Can do more than display string

Have access to student objects


```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public Student removeStudent() {  
        if (studentHeap.isEmpty())  
            System.out.println("Can not remove student from empty queue");  
    }  
}
```

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public Student removeStudent() {  
        if (studentHeap.isEmpty())  
            What to do here?  
  
    }  
}
```

What does your language's collection classes do in this case?

Why should you do the same?

Standard In/Out

Useful in

- Debugging
- Unix commands
- Command line installers
- Command line programs

Not useful in

- GUI applications
- Desktop & Mobile
- Front end web
- Server side applications
- Enterprise
- Embedded software

Java only recently has reasonable ways to read standard in

```
class PriorityQueue {
    private Student[ ] studentHeap;

    public static void main([String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter size of priority queue");
        PriorityQueue waitlist = new PriorityQueue(in.nextInt());

        char ch = 0;

        do {
            System.out.println("\nPriority Queue Operations\n");
            System.out.println("1. Insert student in queue");
            System.out.println("2. Remove student from queue");
            System.out.println("3. Print queue");
            etc.
        }
    }
}
```

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public Student removeStudent() {  
  
  
  
  
  
  
  
  
  
}
```

What does your language's collection call this method?

Why does it matter?

Naming Methods

Look to see if your class library has a similar method

Consider using the same name

- Fewer names to remember

- Programmers already know the meaning

- Start to get replaceable components

```
class PriorityQueue {  
    private Student[ ] studentHeap;  
  
    public Student removeStudent() {  
  
  
  
  
  
  
  
  
  
    }  
}
```

What interface/class does PriorityQueue implement in your language?

Why does it matter?

Java Queue Interface

Implementing classes

AbstractQueue, ArrayBlockingQueue, ArrayDeque, ConcurrentLinkedDeque, ConcurrentLinkedQueue, DelayQueue, LinkedBlockingDeque, LinkedBlockingQueue, LinkedList, LinkedTransferQueue, PriorityBlockingQueue, PriorityQueue, SynchronousQueue

```
public foo(Queue x) {  
    blah  
}
```

```
foo(Z);           // Z can be any of the above classes
```


Program to an Interface, not an Implementation

```
public foo(ArrayList x) {  
    blah  
}
```

```
public foo(List x) {  
    blah  
}
```

ArrayList,
AttributeList,
CopyOnWriteArrayList,
LinkedList,
RoleList,
RoleUnresolvedList,
Stack,
Vector

```
public interface PriorityQueue {  
    void add(Student s);  
  
    Student remove();  
  
    Student peek();  
  
    boolean isEmpty();  
  
    int size();  
  
}
```

How many types of PriorityQueues
that hold students are you going to implement?

So why bother with interface?

```
class PriorityQueue {  
    public void add(String name, String redid, String email, String address,  
                    int units, double gpa) {  
    }  
}
```

Why should a PriorityQueue know how to create a Student object?

```
class PriorityQueue {  
    public void add(Student item) {  
    }  
}
```

What does the add method return in your language's add method?

Why does it matter?

```
class Student {  
    private String name;  
    private String redid;  
    private String email;  
    private int units;  
    private double gpa;  
    private double priority;  
  
    public String getName() { return name;}  
    public String getRedid() { return redid;}  
    public String getEmail() { return email;}  
    public int getUnits() { return units;}  
    public double getGpa() { return gpa;}  
    public double getPriority() {return priority;}  
  
    public void setName(String name) { this.name = name;}  
    etc
```

Improper Inheritance

```
class StudentNode extends Student {  
    private StudentNode parent;  
    private StudentNode leftChild;  
    private StudentNode rightChild;
```

The node in the tree now has a

name

address

email address

etc.

They are not used

```
class StudentNode {  
    private Student data  
    private StudentNode parent;  
    private StudentNode leftChild;  
    private StudentNode rightChild;
```

A StudentNode is not a type of student

A StudentNode has a student

Is-a Has-a Test

If A has a B

B is a field in A

If A is a type of B

A is a subclass of B

```
class Priority {  
    static let instance = Priority()                Not a class, just a function  
  
    func getPriority(student: Student) -> Double {  
        return 0.7 * Double(student.units) + 0.3 * student.gpa  
    }  
}
```

```
let student = Student(name: "Roger", redId: "1234566789",  
                      units: 25, gpa: 3.7, email: "foo@bar.com")  
Priority.instance.getPriority(student: student)
```

```
func getPriority(student: Student) -> Double {  
    return 0.7 * Double(student.units) + 0.3 * student.gpa  
}
```

```
let student = Student(name: "Roger", redId: "1234566789",  
                      units: 25, gpa: 3.7, email: "foo@bar.com")  
getPriority(student: student)
```

Simpler to just use function!

But function just used data from student!

Heuristics

Keep related data and behavior in one place

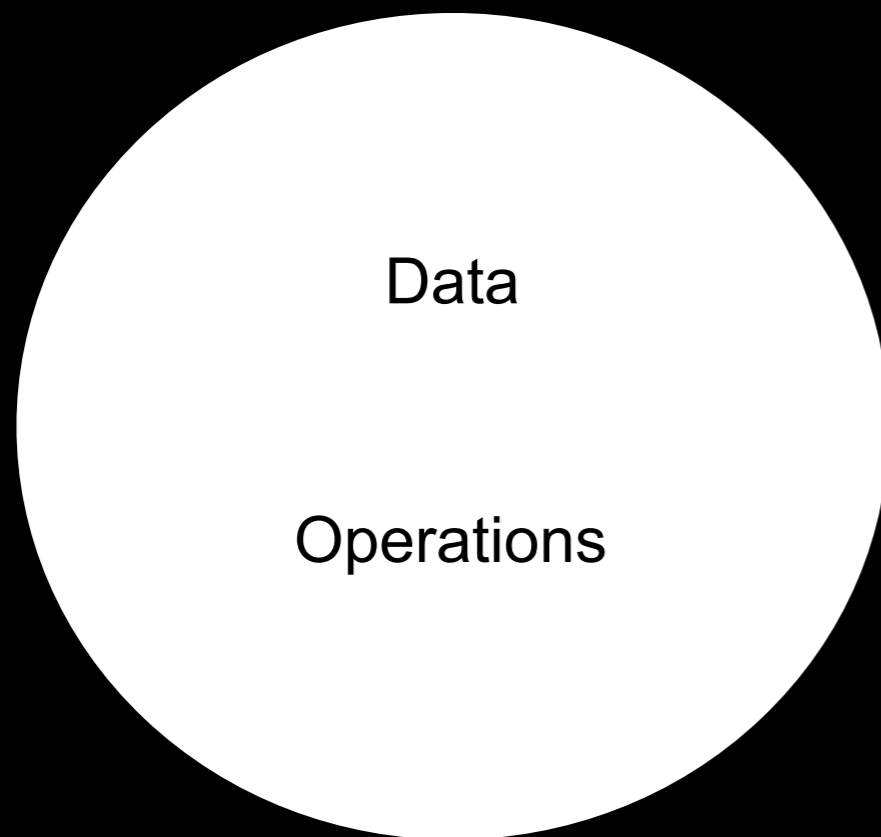
A class should capture one and only one key abstraction

Class

Represents an abstraction

Encapsulates data and operations of the abstraction

Hide design decisions/details



Utility method

Method in class that

- Does not access any field (data member, instance variables)

- Just uses parameters

```
class Student {  
  
    func priority() -> Double {  
        return 0.7 * Double(this.units) + 0.3 * this.gpa  
    }  
}
```

```
let student = Student(name: "Roger", redId: "1234566789",  
                      units: 25, gpa: 3.7, email: "foo@bar.com")  
student.priority()
```

Keeping operations with data

Swift does not prepend get in front of accessors

More Swift Like

```
struct Student {  
  
    var priority: Double {  
        get {  
            return 0.7 * Double(this.units) + 0.3 * this.gpa  
        }  
    }  
}  
  
let student = Student(name: "Roger", redId: "1234566789",  
                      units: 25, gpa: 3.7, email: "foo@bar.com")  
student.priority
```

```
class PriorityQueue {  
  
    public String toString() {  
        printQueue();  
        return "End of current queue";  
    }  
}
```

Information Hiding

```
class PriorityQueue {  
    private ArrayList<Student> studentList = new ArrayList<Student>();  
  
    public ArrayList<Student> print() {  
        for (Student student : studentList) {  
            System.out.println(student.toString());  
        }  
        return studentList;  
    }  
}
```

Anyone can now edit the student list

```
public ArrayList<Student> print() {  
    for (Student student : studentList) {  
        System.out.println(student.toString());  
    }  
    return studentList.copy();  
}
```

```
class PriorityQueue {  
    ArrayList<Student> studentList = new ArrayList<Student>();  
  
    public static Student addStudent() {  
        code not shown  
    }  
}
```

How to add students?

```
PriorityQueue studentQueue = new PriorityQueue();  
studentQueue.studentList.add(new Student());
```