COSC 121

Computer Programming II

Dr. Bowen Hui

University of British Columbia Okanagan

bowen.hui@ubc.ca

Part 1 Solutions

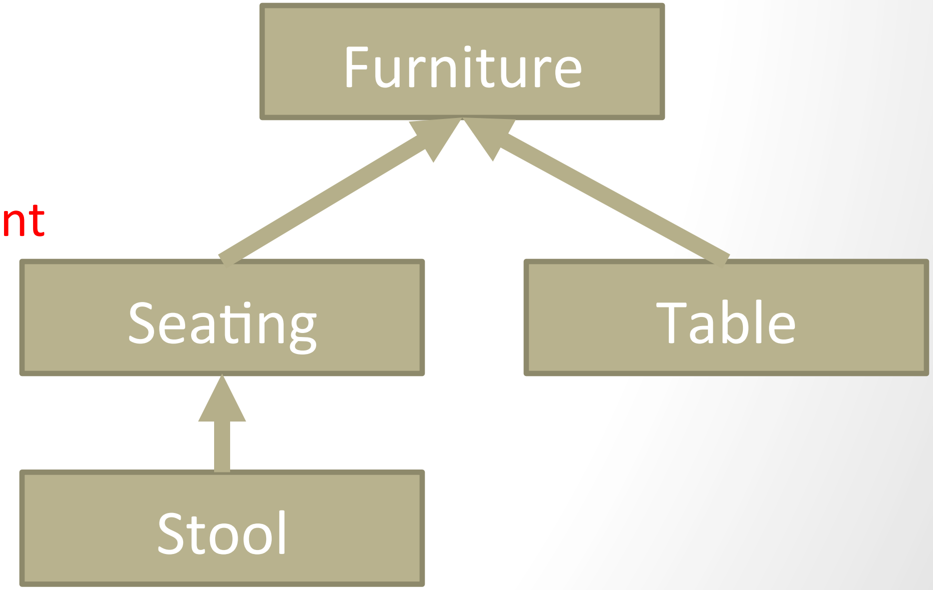
- Average: 7.4/11
- Maximum: 11/11

1. C several people said A or B
2. E many people said D
3. D
4. B typo – exam removed this question, out of total of 49 now
5. D
6. A or B
7. C
8. A
9. D many people said E or A
10. B lots of people said C
11. B

Part 2 Questions 1-2 Solutions

- Average: 9.1/11
- Maximum: 11/11

1. Finding all the is a relationships
   1. Seating is a Furniture
   2. Stool is a Seating
   3. Stool is a Furniture
   4. Table is a Furniture
2. Drawing the relationships
   1. Don’t forget boxes
   2. Arrows need to point to parent



Part 2 Question 3 Sample Solution

3. Making the abstract Shape class

i. Declaring calcArea as int

ii. Having a constructor

```
public abstract class Shape{
     private int numSides;
     private double length;

     public abstract double calcArea();
     public void setLength(int newLen){
           length = newLen;
     }
}
```

Part 2 Question 4 Sample Solution

4. Making the Circle class

i. Syntax errors, Pi, r^2, π, r^2 etc.

ii. Otherwise question was well done.

```
public class Circle extends Shape{
    private double radius;

    public Circle(double r){
        radius = r;
    }
    public double calcArea(){
        return 3.14*radius*radius;
    }
}
```

Part 3 Question 1 Sample Solution

- Average: 8.6/10
- Maximum: 10/10

1. Making animal classes and a Farm class

i. Forgetting to use getters for name

ii. Various mistakes in using the loop and array

```
public class Farm{
        public static void main(String[] args){
                Animal[] pets = new Animal[3];
                pets[0] = new Pig(“Major”, 4);
                pets[1] = new Pig(“Snowball”, 4);
                pets[2] = new Raven(“Moses”, 2);
                for(int i = 0; i < pets.length; i++)
                    System.out.println(pets[i].getName() + “ says “ + pets[i].speak());
        }
}
```

Part 3 Question 2a Sample Solution

- Average: 4.5/10
- Maximum: 10/10

2a. Making the comment class

i. Need to have Comment next as an attribute as the class will be used as part of a list

ii. Need getters and setters for the attribute next

iii. Trying to use super() in the constructor

iv. Need to define abstract methods getAuthor() and getText()

v. In the compare to method concatenate getAuthor() and getText() then compare using compare to.

```
public int compareTo(Object second){
     Comment other = (Comment)second;
     String both = author + text;
     String otherBoth = other.getAuthor() +
                              other.getText();
     return both.compareTo(otherBoth);
}
```

Part 3 Question 2b Sample Solution

- Average: 3.1/8
- Maximum: 8/8

```
public void delete(Comment node){
    if(clist != null){   //check if there are any nodes in the list
        Comment curr = clist;
        Comment prev = null;
        while(curr!=null){
            if(curr.compareTo(node)!=0){ //this is not the node we want
                prev = curr;
                curr = curr.getNext();
            }else{                       //we found it!!!
                if(prev == null)
                    clist = curr.getNext();
                else
                    prev.setNext(curr.getNext());
                break;
            }
        } //end of while
    } //end of if
} //end of method
```