# Unit 4 Discussion: OOP

1. What was the hardest part of Unit 3 assignments? Please explain. (Note: the purpose of this discussion question is for you to reflect on your assignments last week and share some tips <what failed, what worked, …> with the class)

2. What did you find most confusing or difficult about what you read this week? You may also ask for help on a PA/CA or use of zyBook. (Note: be specific. For example, instead of “xx is hard”, identify the topic and illustrate with an example.)

3. Draw a UML class diagram for the TaxTableTools.java in zyDE 7.21.4. Please follow the notation we introduced in the lecture.

4. You may add the keyword static in the place of ? in line \_\_\_\_\_\_\_\_ in the following code:

1 public class Test {   
2 private int age;  
3   
4 public ? int square(int n) {   
5 return n \* n;  
6 }  
7  
8 public ? int getAge() {  
9 return age;   
10 }  
11 } // end class Test

A. in line 4  
B. in line 8  
C. in both line 4 and line 8  
D. none

5. Which of the following statements are true? (multiple answers)

A. Local variables do not have default values.  
B. Data fields (data members in a class) have default values.  
C. A variable of a primitive type holds a value of the primitive type.  
D. A local variable of a class type will be assigned null automatically.   
E. You may assign an int value to a reference variable.

6. Analyze the following code.

public class Test {

public int x;

public Test(String t) {

System.out.println("Test");

}

public static void main(String[] args) {

Test test = new Test();

System.out.println(test.x);

}// end main

} // end class Test

A. The program has a compilation error because System.out.println method cannot be invoked from

the constructor.

B. The program has a compilation error because x has not been initialized.

C. The program has a compilation error because you cannot create an object from the class that

defines the object.

D. The program has a compilation error because Test does not have a default constructor.

E. none of the above

7. What is the output for the following code? Show all output precisely in the order they are generated. For each print, indicate from which method it’s from and which variable (indicate instance variable, parameter, or local variable) it’s printing.

public class TraceClass {

private int num;

public TraceClass() {

System.out.println(num);

}

public TraceClass(int num) {

this();

this.num += 50;

System.out.println(num);

}

public void doIt() {

System.out.println(num);

}

public void doIt(int num) {

this.num = 300;

doIt();

System.out.println(num);

}

public static void main(String[] args) {

int num = 10;

TraceClass tc = new TraceClass(num);

System.out.println(num);

tc.doIt(100);

} // end main

} // end TraceClass class

/\* TestPoint (client) and MyPoint are referenced in the next few questions.   
 TestPoint should be put in a file named TestPoint.java  
 MyPoint should be put in a file named MyPoint.java. As an alternative MyPoint may be put in the same file TestPoint   
resides, after TestPoint and with the public modifier on the class heading taken out (i.e. change from public class MyPoint to class MyPoint)  
\*/  
public class TestPoint {  
 public static void main(String[] args) {  
 // create two MyPoint objects  
 MyPoint p1 = new MyPoint();  
 MyPoint p2 = new MyPoint();

// set coordinates  
 p1.setX(5);  
 p1.setY(10);

// ADD CODE HERE

} // end main

} // end class TestPoint

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// a MyPoint object represents a pair of (x, y) coordinators  
public class MyPoint {  
 private int x;  
 private int y;  
  
 public void setX(int x) {

this.x = x;

}  
  
 public void setY(int y) {  
 this.y = y;

}  
  
 public int getX() {

return x;

}  
  
 public int getY() {

return y;

}  
   
} // end class MyPoint

8. Given the MyPoint class and TestPoint class (client) listed above, add two new instance methods to the MyPoint class:

// set this point's (x, y) coordinates to the new values   
 public void setLocation(int newX, int newY)  
  
 // return a String representation of the calling object in the format of [x=4,y=3]  
 public String toString()

Your two new instance methods should work with those three new lines of code added into the main():

p2.setLocation(3, 2); // set p2 to x as 3 and y as 2

// print each point.

// toString() of a class will be called when its object is used in print/ln/f

// or concatenated with a string   
System.out.println("p1 is: " + p1); // should print p1 is: [x=5,y=10]  
System.out.println("p2 is: " + p2); // should print p2 is: [x=3,y=2]

9. Given the MyPoint class and TestPoint class (client) listed above, add two new instance methods to the MyPoint class:

// returns the distance between this Point and (0, 0): d = (x \* x + y \* y)1/2  
// use Math.sqrt() to calculate square root of a number  
public double distanceFromOrigin()  
  
// shifts this point's location by the given amount.  
// afterwards this point should be at location x+dx, y+dy  
public void translate(int dx, int dy)

Your two new instance methods should work with those new lines of code added into the main() :

// set point p2  
p2.setX(3);  
p2.setY(2);  
  
// print distance from p1 origin (0, 0)   
System.out.println("p1 distance from p2 - " + p1.distanceFrom(p2));  
  
// translate to new locations and print again  
p1.translate(11, 6);  
p2.translate(1, 7);  
System.out.println("Now p1 is at: (" + p1.getX() + ", " + p1.getY() + ")");  
System.out.println("Now p2 is at: (" + p2.getX() + ", " + p2.getY() + ")");

10. Given the MyPoint class and TestPoint class (client) listed above, add one new instance method to the MyPoint class:

// returns the distance between this Point and the parameter Point.

// the distance between two points (x1, y1) and (x1, y2) may be calculated:

// d = ( (x1 – x2)2 + (y1 – y2)2 )1/2  
// use Math.sqrt() to calculate square root of a number  
public double distanceFrom(MyPoint p)

Your new instance method should work with those new lines of code added into the main():

p1.setX(6);  
p1.setY(-2);

// set point p2  
p2.setX(3);  
p2.setY(2);  
  
// print distance from p1 (6, -2) to p2 (3, 2)   
System.out.println("p1 distance from p2: " + p1.distanceFrom(p2));

// result should be 5.0

11. Given the MyPoint class and TestPoint class (client) listed above, add one new instance method to the MyPoint class:

// returns true if this Point and the parameter Point identify the same

// location  
public boolean equals(MyPoint p)

Your new instance method should work with those new lines of code added into the main():

p1.setX(6);  
p1.setY(-2);

// set point p2  
p2.setX(3);  
p2.setY(2);  
  
// should print false   
System.out.println( p1.equals(p2) );

p2.setX(p1.getX());  
p2.setY(p2.getY());  
// should print true   
System.out.println( p1.equals(p2) );

12. Given the MyPoint class and TestPoint class (client) listed above, add two constructors to the MyPoint class, a default constructor and a two-parameter constructor. The constructors should work with those new lines of code added into the main():

MyPoint p3 = new MyPoint(); // new MyPoint object at location (0, 0)

MyPoint p4 = new MyPoint(3, 4); // new MyPoint object at location (3, 4)

13. Given the MyPoint class and TestPoint class (client) listed above, add to the MyPoint class a private static int data member called numOfPoints and a public static getNumOfPoints() to return the current value of this static member. Assume the two constructors listed in the last questions already exists. Revise them so numOfConostructors is incremented for each newly created MyPoint object. The addition/update should work with those new lines of code added into the main():

MyPoint p3 = new MyPoint(); // new MyPoint object at location (0, 0)

MyPoint p4 = new MyPoint(3, 4); // new MyPoint object at location (3, 4)

System.out.println( MyPoint.getNumOfPoints() ); // should print # of MyPoint objects created so far

14. Does the String class have any accessor methods? Name 2 accessor methods if any and explain the source code of those two accessor methods line by line. <http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/687fd7c7986d/src/share/classes/java/lang/String.java>

15. Does the String class have any mutator methods? Name 2 if any and explain the source code of those two mutator methods line by line. <http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/687fd7c7986d/src/share/classes/java/lang/String.java>

16. Read source code of ArrayList and answer the following questions (pick one). <http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/687fd7c7986d/src/share/classes/java/util/ArrayList.java>

16.a. What’re the purposes of the instance data member size, internally and externally?

16.b. When are elementData.length used (provide a couple of examples) and for what reasons?

16.c. Explain line by line how get(int) works (line 413), including methods that are invoked by get() (line #, and also line by line explanation), as long as those are methods of ArrayList.

16.d. Explain line by line how set(int, E) works (line 428), including methods that are invoked by set() (line #, and also line by line explanation), as long as those are methods of ArrayList.

16.e. What’s the relationship between contains(Object) (line 284) and indexOf(Object) (line 295)? What’s the benefit of providing both methods?

16.f. How do add(E) and add(int, E) (line 442, 457) work similarly and differently? How does the implementation differ?

16.g. How do remove(int) and remove(Object) (line 476, 504) work similarly and differently? How does the implementation differ?