CS1102: Data Structures and Algorithms

Tutorial 1 – Java and Problem Solving

(28, 29 January 2010)

1. [**Concept recap**] What is the difference between the following keywords and concepts? Please provide examples to help to explain.

   a) `throw` and `throws`

   b) ArrayList and Vector

   c) Auto-boxing and auto-unboxing. Which one of lines 1, 2, 3, and 4 are examples of auto-boxing and auto-unboxing?

```
import java.util.*;

// Prints a frequency table of the words on the command line
public class Frequency {
   public static void main(String[] args) {
      HashMap<String, Integer> m = new HashMap<String, Integer>();//1
      for (String word : args) {
          Integer freq = m.get(word);                              //2
          m.put(word, (freq == null ? 1 : freq + 1));              //3
          System.out.println(freq);                                //4
      }
      System.out.println(m);
   }
}
```

2. [**Common issues**] What is the output of the following code?

   a)
```
int x = 1;
Integer y = 1;
x = y++ + ++y;
System.out.println(x);
int z = x > y++ + ++y ? x++ + ++x : ++x + y;

System.out.println(x);
System.out.println(y);
System.out.println(z);
```

   b)
```
Integer eye = 42;
Double d = 42.0;
int i = 42;
double dd = 42.0;
System.out.println(i == eye);
System.out.println(i == d);
System.out.println(eye == dd);
System.out.println(d == dd);
System.out.println(eye.equals(d));
```

c)
```
int []array = {1,2,3,4};
for ( int i : array ) {
   array[i] = 0;
}
for ( int i : array ) {
   System.out.print(i + " ");
}
```

3. [**Static, scope**] Consider the following program. pack.java is stored in a directory *packed*, which has the same name as the package. test.java is stored in the directory that contains *packed*.

```
// ./packed/pack.java
package packed;
public class pack
{
     public int x1 = 1;
     protected static int x2 = 2;
     int x3 = 3;
     private int x4 = 4;
}
// end of file

// ./test.java
import packed.pack;
class test {
     private int x1 = 1;
     static int x2 = 2;

     public static void main( String args[] )
     {
          pack p = new pack();
          System.out.println( p.x1 );            //1
          System.out.println( p.x2 );            //2
          System.out.println( p.x3 );            //3
          System.out.println( p.x4 );            //4
          test t = new test();
          test t1 = new test();
          t.printSum(t1, p);
     }
     void printSum(test t, pack p)
     {
          System.out.println(this.x1 + t.x1);    //5
          System.out.println(this.x2 + p.x2);    //6
          System.out.println(test.x2 + pack.x2); //7
     }
}
// end of file
```

Which ones of the seven marked lines are illegal? Why?

4. [**Generics**] Consider the following classes:

```
public class AnimalHouse<E> {
     private E animal;

     public void setAnimal(E x) {
          animal = x;
     }

     public E getAnimal() {
          return animal;
     }
}

public class Animal{
}

public class Cat extends Animal {
}

public class Dog extends Animal {
}
```

For the following code snippets, identify whether the code fails to compile, compiles with a warning, generates an error at runtime, or none of the above. If there is an error or warning, explain why.

a)
```
AnimalHouse<Animal> house = new AnimalHouse<Cat>();
```

b)
```
AnimalHouse<Dog> house = new AnimalHouse<Animal>();
```

c)
```
AnimalHouse house = new AnimalHouse();
house.setAnimal(new Dog());
```

d)
```
AnimalHouse<?> house = new AnimalHouse<Cat>();
house.setAnimal(new Cat());
```

e)
```
AnimalHouse<Cat> house = new AnimalHouse<Cat>();
house.setAnimal(new Cat());
```