Multiple choice

6) Which of the following statements are true of Java programs?
   a) A class can implement more than one interface.
   b) A class can extend more than one class.
   c) Multiple inheritance is a built-in part of the language.
   d) (b) & (c)
   e) None of the above statements are true.

7) Which of the following statements are true about Java programs?
   a. Overriding refers to methods with the same name & parameter list but have separate definitions in the parent and child class.
   b. Overloading refers to methods that have the same name but are distinguished by their parameter lists.
   c. Preceding an attribute with the word private in a class definition is an example of information hiding.
   d. Java programs are both compiled and interpreted.
   e. All of the above are true

Extra practice questions for you try on your own

1)   For this question refer to class X
```
public class X
{
    public void fun () { System.out.println("This is fun"); }
}
```

   a. Write the definition for class Y that so it's a child of X.

   b. Write an overloaded version of method fun that takes an integer as parameter and displays it onscreen.

   c. Write an overridden version of method fun that takes an integer as a parameter, doubles it and then displays the doubled value onscreen.

2) Trace the output of the following program.

JT's note: parts of this question are fairly easy, parts of it start getting harder..don't get too frazzled by the harder parts. This question just involves concepts that you've already seen: inheritance, overloading, overriding, shadowing. My final hints: (1) you should keep in mind that there are two separate objects (an 'A' object and a 'B' object) (2) as you trace the program write or draw out what is happening each step of the way (don't try to hold it all in your head or you'll likely make many mistakes).

```
public class DriverTrace
{
  public static void main (String [] args)
  {
    A a = new A ();
    B b = new B ();
```

```java
        System.out.println(a);
        System.out.println(b);
        a.fun();
        b.fun(123);
        System.out.println(a);
        System.out.println(b);
        a.fun(b.z);
        b.fun(a.y);
        System.out.println(a);
        System.out.println(b);
    }
}

public class A
{
    protected int x = 1;
    protected int y = 2;
    protected void fun () { x = x + 1; }
    protected void fun (int a) { y = y + a; }
    public String toString ()
    {
        String s = "";
        s = x + " " + y;
        return s;
    }
}

public class B extends A
{
    protected int x = 10;
    protected int z = 20;
    protected void fun ()
    {
        super.fun(z);
    }
    protected void fun (int a)
    {
        super.fun(a);
    }

    public String toString ()
    {
        String s = "";
        s = s + super.x + " " + y + " ";
        s = s + x + " " + z;
        return s;
    }
}
```