Exercises 7

1) Consider the following classes:

```java
public class Vehicle {...}
public class Car extends Vehicle {...}
public class SUV extends Car {...}
```

Which of the following are legal statements? Explain why any statement is invalid.

- a. Car c = new Vehicle();
- b. Vehicle v = new Car();
- c. SUV s = new SUV();
- d. SUV s = new Car();
- e. Vehicle v = new SUV();
- f. Car c = new SUV();

2) Consider the following classes:

```java
public class First {
    public void method2() {
        System.out.println("First2");
    }

    public void method3() {
        method2();
    }
}

public class Second extends First {
    public void method2() {
        System.out.println("Second2");
    }
}

public class Third extends Second {
    public void method1() {
        System.out.println("Third1");
        super.method2();
    }

    public void method2() {
        System.out.println("Third2");
    }
}
public class Fourth extends First {
    public void method1() {
        System.out.println("Fourth1");
    }

    public void method2() {
        System.out.println("Fourth2");
    }
}

Suppose the following variables are defined:

First var1 = new Second();
First var2 = new Third();
First var3 = new Fourth();
Second var4 = new Third();
Object var5 = new Fourth();
Object var6 = new Second();

Indicate below the output that would be produced by each statement shown. If the statement causes an error, write the word error to indicate this and explain why.

a) var1.method2();

b) var2.method2();

c) var3.method2();

d) var4.method2();

e) var5.method2();

f) var6.method2();

g) var1.method3();

h) var2.method3();

i) var5.method3();

j) var6.method3();

k) ((First) var5).method3();

l) ((Second) var6).method1();
3) A company has written a large class `BankAccount` with many methods including

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public BankAccount(Startup s)</code></td>
<td>Constructs a BankAccount object using information in s</td>
</tr>
<tr>
<td><code>public void debit(Debit d)</code></td>
<td>Records the given debit</td>
</tr>
<tr>
<td><code>public void credit(Credit c)</code></td>
<td>Records the given credit</td>
</tr>
<tr>
<td><code>public int getBalance()</code></td>
<td>Returns the current balance in pennies</td>
</tr>
</tbody>
</table>

The `BankAccount` class constructor sets the initial balance on the basis of the startup information. Assume that only debits and credits change an account's balance.

Design a new class `MinMaxAccount` whose instances can be used in place of a `BankAccount` object, but include new behaviors of remembering the minimum and maximum balances ever recorded for the account. The class should have a constructor that accepts a `Startup` parameter.

Include these new methods in your class

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public int getMin()</code></td>
<td>Returns the minimum balance in pennies.</td>
</tr>
<tr>
<td><code>public int getMax()</code></td>
<td>Returns the maximum balance in pennies.</td>
</tr>
</tbody>
</table>