class Plant {
    public void grow() {
        System.out.println("--<>--|");
    }
    public void photosynthesize() {
        System.out.println("nom nom nom");
    }
}

class Flower extends Plant {
    @Override
    public void grow() {
        System.out.println("*--<>--|");
    }
    public Flower pollinate() {
        return new Flower();
    }
}

class Rose extends Flower {
    @Override
    public void grow() {
        System.out.println("@--<>--|");
    }
    @Override
    public Flower pollinate() {
        return new Rose();
    }
    public void prick() {
        System.out.println("Ouch!");
    }
}
Consider the following code segment.

```java
Object o = new Rose();
Plant p1 = new Plant();
Plant p2 = new Flower();
Plant p3 = new Rose();
Rose r = (Rose) o;
// new code goes here
```

1. Write the static type and dynamic class for each of these variables based on the declarations given above.

<table>
<thead>
<tr>
<th>variable</th>
<th>static type</th>
<th>dynamic class</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>Object</td>
<td>Rose</td>
</tr>
<tr>
<td>p1</td>
<td>Plant</td>
<td>Plant</td>
</tr>
<tr>
<td>p2</td>
<td>Flower</td>
<td>Flower</td>
</tr>
<tr>
<td>p3</td>
<td>Rose</td>
<td>Rose</td>
</tr>
<tr>
<td>r</td>
<td>Rose</td>
<td>Rose</td>
</tr>
</tbody>
</table>

2. Suppose that each of the following lines of code were inserted into the code segment at the point indicated. Indicate whether the line compiles and whether it completes without throwing an exception; if it does, write the value of the expression or the output printed to the console.

(a) o.grow(); □ Compiles □ No exception ⇒
(b) p2.grow(); □ Compiles □ No exception ⇒
(c) r.grow(); □ Compiles □ No exception ⇒
(d) p3.prick(); □ Compiles □ No exception ⇒
(e) r.prick(); □ Compiles □ No exception ⇒
(f) ((Rose) p2).prick(); □ Compiles □ No exception ⇒
(g) ((Object) p2).prick(); □ Compiles □ No exception ⇒
(h) r.pollinate().prick(); □ Compiles □ No exception ⇒
(i) o instanceof Rose □ Compiles □ No exception ⇒
(j) p2 instanceof Rose □ Compiles □ No exception ⇒
(k) r instanceof String □ Compiles □ No exception ⇒
(l) p1.photosynthesize(); □ Compiles □ No exception ⇒
(m) p2.photosynthesize(); □ Compiles □ No exception ⇒
(n) p3.photosynthesize(); □ Compiles □ No exception ⇒

3. In the remaining space, examine the merits of the Java programming language and contrast them with the features of OCaml. Structure your response in the form of a justificatory essay, being sure to include at least one counter-argument and rebuttal. Your essay should be succinct but persuasive, and no longer than 500 words in length.