1. True or False:

(a) T Everything in an Alice world is an object.
(b) F Methods cannot call other methods.
(c) T There are many predefined classes in Alice that the programmer can choose from.
(d) F A variable can be used anywhere in the program, even before the declaration.
(e) F The random number generation function only produces numbers between 0 and 1.
(f) F Only one or two parameters can be accepted by a method.
(g) F Comments in your program tell the computer how to run your program.
(h) F Both portions of an If/Else statement must contain statements.
(i) T An If/Else statement can be included in either part of another If/Else statement.
(j) F The loop never stops if the condition remains false.

2. (a) Explain what a loop statement does:

A loop statement repeats a set of statements a fixed number of times, or while some condition holds.

(b) What is an infinite loop? Give an example.

An infinite loop is a loop that never stops. For example, a loop that starts "while true" will continue forever since the condition is always true.

3. Write the Alice commands that will modify the cow object to be 50% transparent, blue, and riding in the helicopter object (ie set the vehicle to helicopter).

<table>
<thead>
<tr>
<th>Properties</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>cow.opacity = 50%</td>
</tr>
<tr>
<td>opacity</td>
<td>cow.color = blue</td>
</tr>
<tr>
<td>vehicle</td>
<td>cow.vehicle = helicopter</td>
</tr>
</tbody>
</table>

4. To the right of each line of code, indicate the value of the logical expression after those lines have been executed.
### (a) Continue = true  
Stop = false

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>!Stop</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (b) a = -1, b = 2, c = 5

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c - 2) == 0</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### (c) (no change)

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a != 0</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (d) Increment a by 1

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a == 0</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (e) Increment b by 3

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>b == c</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (f) Set Value of a to b+c

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a == 0) AND Stop</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### (g) Set Value of Stop to true

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>!Stop OR Stop</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (h) (no change)

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>!Continue AND Stop</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### (i) (no change)

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a ≥ b/2) AND (c ≥ b)</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

### (j) (no change)

<table>
<thead>
<tr>
<th>expression</th>
<th>True or False?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IEEE Remainder of c/2) == 0</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

---

### 5. Assume that the object frog has a method called hop with parameter distance that controls how far forward the frog hops. Write the my first method that makes the frog hop 1 meter, 2 meters, and then 3 meters.

**my first method:**

```java
frog.hop(1 meter)
frog.hop(2 meters)
frog.hop(3 meters)
```
6. Write the my first method for a world that shows an astronaut bounding across the lunar landscape, then planting a flag, turning around, and heading back to his initial position.

In addition, assume that the following methods have already been written for you:

```plaintext
astronaut.bound()
astronaut.plantFlag()
```

my first method:

```plaintext
astronaut.bound()
astronaut.plantFlag()
astronaut.turn(left, 1/2 revolution)
astronaut.bound()
```

7. (a) Write an If/Else statement that causes an object called bunny to double in size 30% of the time.

```
If (choose true 0.3 (30%) of the time
    bunny.resize(2)
```

(b) Write an If/Else statement that causes an object called bunny to turn red if it is within 2 meters of an object called stove, otherwise the bunny should turn blue

```
If (bunny.isWithin(2meters, stove)
    bunny.color = red
Else
    bunny.color = blue
```

8. In words, what does the following loop do?

```plaintext
world.my first method ()
// Programming Project 3.5
Do together
While (both (( frog distance in front of square asSeenBy = ground ) > 0.25 )
    and (( frog2 distance in front of square asSeenBy = ground ) > 0.25 ) )
Do together
    camera move amount = 1 meter toward target = stop style = abruptly duration = 1.5 seconds
frog.hop ( random number minimum = 0.25 maximum = 2.25 )
frog2.hop ( random number minimum = 0.25 maximum = 2.25 )
```

The frogs hop a random amount toward the square, until one is within a 1/4 of a meter of the square, and then they stop. The camera follows the frogs as they race towards the finish.
9. Write the **my first method** which contains a **Loop** statement that causes an object **skater** to spin in place 10 times. You may assume that the method **skater.spin** already exists.

   **my first method:**

   Loop 10 times
   
   skater.spin();

10. Write the method **DeliverCrate** method for the object **Blimp**. Your method should check to see if the Blimp is directly above the **Boat** using the **is Above** function. If it is, an object **Crate** should fall from the blimp to the boat (it may miss). The crate then ”disappears” (ie becomes invisible and returns to the blimp). This should be repeated forever.

   (Suggestions: You may assume that the blimp is always 50 meters from the ocean. Further, making the blimp the “vehicle” for the crate will make it travel with the blimp.)

   **DeliverCrate()**

   crate.vehicle = blimp
   while true
     if ( blimp.isAbove(Boat) )
       crate.opacity = 100%;
       crate.move(down, 50 meters, 2 seconds)
       crate.opacity = 0%;
       crate.move(up, 50 meters)