Given below are the world functions.

- **Boolean logic**
  - `not a`
  - `both a and b`
  - `either a or b`, or both

- **Math**
  - `a + b`
  - `a - b`
  - `a > b`
  - `a < b`
  - `a <= b`

- **Random**
  - `choose true` probability of True of the time
  - `random number`

- **String**
  - `a joined with b`
  - `what as a string`

- **Ask user**
  - `ask user for a number`
  - `ask user for yes or no`

- **Advanced math**
  - `minimum of a and b`
  - `maximum of a and b`
  - `absolute value of a`

- **Time**
  - `time elapsed`
  - `hour of AM or PM`
  - `hour of day`
  - `minute of hour`
  - `second of minute`

- **Miscellaneous**
  - `square root of a`
  - `floor a`
  - `ceil a`
  - `round a`
  - `a converted from radians to degrees`
  - `a converted from degrees to radians`
  - `right, up, forward`
Given below are the chicken properties and methods.

### Chicken’s Details

<table>
<thead>
<tr>
<th>Properties</th>
<th>Methods</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>opacity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vehicle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>skin texture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fillingStyle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pointOfView</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isShowing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Methods**
  - move
  - turn
  - roll
  - resize
  - say
  - think
  - playSound
  - moveTo
  - moveToward
  - moveAwayFrom
  - orientTo
  - turnToFace
  - pointAt
  - setPointOfView
  - setPose
  - standUp
  - moveAtSpeed
  - turnAtSpeed
  - rollAtSpeed
  - constrainToFace
  - constrainToPointAt
Given below are the chicken functions.

Tiles at the bottom of the Alice window.
1. (3 pts) Consider the following html code.

```html
<html>
<head>
<title>Football at Duke</title>
</head>
<body>
<h2>Next Home Game</h2>
<h1>Sat. Oct. 18, 2008!</h1>
<p>Get your tickets now!</p>
</body>
</html>
```

Which picture corresponds to this code?

A) ![A Image](image1.png)

B) ![B Image](image2.png)

C) ![C Image](image3.png)

D) ![D Image](image4.png)
2. (4 pts) Consider the following html code that is part of an .html page.

```
<img src="http://www.cs.duke.edu/courses/fall08/cps004/skater.gif">

<img src="FairFun.JPG">
```

a) Explain what this code does, that is, what will appear on the web page in place of this code.

b) Explain the differences between the two lines of code above.

3. (3 pts) What do you know about the fall08 part of the URL
http://www.cs.duke.edu/courses/fall08/cps004/ ?

A) fall08 is a domain name

B) fall08 is part of an email address

C) fall08 is a folder

D) fall08 is a part of a machine name
4. (3 pts) Consider the following html code on a web page about things to do at the NC State Fair coming up October 16-26, 2008.

```
<ul>
  <li>rides</li>
  <li>Milk a cow</li>
</ul>
<ol>
  <li>tractor pull</li>
  <li>pig race</li>
</ol>
```

Which one of the following is the portion of the html page produced by this html?

A) • rides  
   • Milk a cow
      1. tractor pull
      2. pig race

B) 1. rides  
    2. Milk a cow  
      • tractor pull  
      • pig race

C) • rides  
   • Milk a cow
      1. tractor pull
      2. pig race

D) 1. rides  
    2. Milk a cow  
      • tractor pull
      • pig race

5. (3 pts) Suppose you drop an object into an Alice world and you cannot see it anywhere. Explain how you can get this object into view so it can be seen.
6. (3 pts) Consider the following html code.

```html
<table border=1>
  <tr> <td>Betty</td> <td>Holberton</td> </tr>
  <tr> <td>Jean</td> <td>Bartik</td> </tr>
  <tr> <td>Kathleen</td> <td>Antonelli</td> </tr>
  <tr> <td>Marilyn</td> <td>Meltzer</td> </tr>
  <tr> <td>Ruth</td> <td>Teitelbaum</td> </tr>
  <tr> <td>Frances</td> <td>Spence</td> </tr>
</table>
```

Which picture corresponds to this code?

A) 

| Betty Holberton | Jean Bartik |
|----------------|
| Kathleen Antonelli | Marilyn Meltzer |
| Ruth Teitelbaum | Frances Spence |

B) 

| Betty Holberton | Jean Bartik | Kathleen Antonelli |
|----------------|
| Marilyn Meltzer | Ruth Teitelbaum | Frances Spence |

C) 

| Betty Holberton | Jean Bartik |
|----------------|
| Kathleen Antonelli | Marilyn Meltzer |
| Ruth Teitelbaum | Frances Spence |

D) 

| Betty Holberton | Jean Bartik | Kathleen Antonelli |
|----------------|
| Marilyn Meltzer | Ruth Teitelbaum | Frances Spence |

7. (3 pts) Which one of the following is a default in Alice (that is, it can happen without you adding it into an Alice world)?

A) Comments

B) Do In Order

C) Do Together

D) Wait
8. (14 pts) Consider the following Alice code in which the lines are numbered.

A) In line 1, is “hello” an argument or a parameter?

B) In line 2, what type of value does “which” return?

C) In line 3, list the words that are arguments.

D) In line 3, list the words that are parameters.

E) In line 3, what type of value does “something” return?

F) Explain what must be true for line 6 to be executed when this program runs.

G) Name one method above that is a user-built class method?
9. (4 pts) Consider the following world that has the three objects: hare, penguin and flamingo (shown below from left to right), and given code. The world has been setup as shown below. The penguin is exactly 1.0 meter from the hare, and the penguin is exactly 1.0 meter from the flamingo.

![Diagram of the world with objects and code]

The diagram below is looking from above over the scene. The hare is represented by the square, the penguin is represented by the circle, and the flamingo is represented by the hourglass. Using the diagram below, draw the path of hare as a solid line and the path of flamingo as a dashed line.
10. (6 pts) Consider the following world.Mystery function.

A) What does `world.Mystery` return when the following call is made?

```
world.mystery num1 = 12 | num2 = 7 | num3 = 20
```

B) What does `world.Mystery` return when the following call is made?

```
world.mystery num1 = 7 | num2 = 4 | num3 = 3
```

C) What does `world.Mystery` return when the following call is made?

```
world.mystery num1 = 1 | num2 = 2 | num3 = 2
```
11. (10 pts) Consider the following Alice world that has three objects: tortoise, cow and chicken.

The world starts as shown in the leftmost figure and finishes as shown in the rightmost figure. Write the Alice code to do the following steps in order.

a) Make sure the chicken stays on the cow when the cow moves in parts b), c) and d).
b) The cow (and thus the chicken too) should turn to face the tortoise.
c) The cow (and thus the chicken too) should move to 2 meters in front of the tortoise.
d) The cow (and thus the chicken) should jump over the tortoise, clearing the tortoise.
e) Finally, the cow takes off without the chicken for 20 meters straight ahead. The chicken then moves to the ground (1 meter down).

Write your code here or on the next page.
12. (8 pts) Complete the following class method called `scare` whose header is shown below. This method has two parameters, an object named “person,” and a number named “value.” This method first has the skeleton turn to face the person and then at the same time, the skeleton says “Boo” and turns its head around “value” times.

For example, in the first call above, the skeleton turns to face Bob, and then at the same time says “Boo” and turns his head around twice. In the second call above, the skeleton turns to face toySoldier, and then at the same time says “Boo” and turns his head around three times.
13. (8 pts) Complete the following function called \texttt{whichOneNotVehicled} that has three parameters, an object named \texttt{ball1}, an object named \texttt{ball2}, and an object named \texttt{transport}, and returns the object that is not vehicled to the transport. You can assume that one of ball1 and ball2 has its vehicle property set to transport and the other one’s vehicle property is not set to transport.

A) For example, in the picture above, the football’s vehicle is not set to car, and the basketball’s vehicle is set to car. Thus the call above returns football.

\begin{verbatim}
world.whichOneNotVehicled ball1 = basketball \quad ball2 = football \quad transport = car
\end{verbatim}

B) Using the objects car, football and basketball, give the Alice code to first have the car move forward ten meters, and second to have the ball that is not vehicled to the car to move to the ground (about 2 meters). You must call the function you wrote in Part A) to receive full credit.

\begin{verbatim}
world.whichOneNotVehicled ball1, ball2, transport
\end{verbatim}
14. (13 pts) Consider that an Alice world has several rectangular objects and that we want to be able to calculate the volume of the objects. The volume is defined as the height * width * depth of an object. For example in the figure below, suppose the tv (the object on the rightside in the figure) has dimensions of 2 meters in height, 3 meters in width and 1 meter in depth. The volume of the tv would be 6, obtained by (2 * 3 * 1).

A) (5 pts) Complete the following function called **volume** that returns the volume of an object. This function has one object parameter named **item** and returns the volume of item.

```plaintext
world.volume
world.volume (Obj item)
```

B) (8 pts) Complete the following function called **LargestVolume** that returns the largest volume of three objects. This function has three object parameters named **box1**, **box2** and **box3**, and returns the volume of the object with the largest volume. You can assume that the three objects have different volumes. You MUST call the function **volume** that you wrote in Part A. You may assume that **volume** works correctly, regardless of what you wrote in Part A.

```plaintext
world.largestVolume
world.largestVolume (Obj box1, Obj box2, Obj box3)
```