1-hr Test #2  April 12 (Mon)

Expectations

1. be able to read and write nested for-loops
2. be able to write code to swap two variables
3. be able to trace the progress of bubble sort, selection sort, and insertion sort for an array with more than 5 items
4. be able to write switch statements and write equivalent if-statements (don't forget putting break at the correct places!)
5. be able to distinguish and declare local and timeline variables
6. be able to determine when you should use local variables, instead of timeline variables, in a function
7. be able to write code to create arrays (one- and multi-dimensional)
8. know the syntax and usages that have been covered:
   a. Movieclip's properties: x- and y-coordinates, % scale, width and height, rotation, opacity
   b. control structures: if-statements, switch statements, for-loop, while loop
   c. keyboard event listener code
   d. mouse event listener code
   e. frame event listener code
9. be able to write code for simultaneous key presses
10. be able to write class definitions, given a diagram showing the inheritance relationships, properties, and methods
11. be able to write code to instantiate a class to create objects and invoke the class methods for the object
12. polymorphism: be able to predict the traced output from the overridden methods, and know how to implement it
13. be able to use the keyword super to invoke the constructor of the superclass and the superclass's methods
14. know the implication and syntax of public, protected, private, and static properties and class methods
15. be able to write code to assign a movieclip as a visual representation for objects instantiated from classes
16. be able to write code to perform the following tasks:
   a. toggle
   b. shuffle a list of items, randomly pick multiple item from an array (each item is picked no more than once)
   c. drag
   d. assign mask to a movieclip instance
Practice Questions

Q1. What are the messages displayed in the Output panel when the following code segment is executed?
   for (var i:int = 0; i < 4; i++)
   {
      for (var j:int = 0; j < 3; j++)
      {
         trace(i + " " + j);
      }
   }

Q2. Write code to swap two variables, a and b.
   (Declare the necessary variables. You can choose a data type.)

Q3. Trace the sorting process (sort in ascending order) of the following array using bubble sort, selection sort, and insertion sort:
   [43, 65, 21, 89, 49, 72]

Q4. Write a switch statement that does the same as the following if-statement.

   if (a == 1)
   {
      trace("one");
   }
   else if (a == 2 || a == 3)
   {
      trace("two or three");
   }
   else if (a == 4)
   {
      trace("four");
   }
   else
   {
      trace("zero");
   }

Q5. Write an if-statement that does the same as the following switch statement.

   switch (a)
   {
      case 1:
         trace("one");
         break;
      case 2:
         trace("two");
      case 3:
         trace("three");
         break;
      case 4:
         trace("four");
         break;
      default:
         trace("zero");
   }
Q6. Write a switch statement that does the same as the following if-statement.

```java
if (a == 1 || a == 3 || a == 7 || a == 17 || a == 79) {
    trace("my favorite number");
}
else if (a == 0 || a == 13 || a == 67 || a == 99) {
    trace("not my favorite number");
}
else {
    trace("just a number to me");
}
```

Q7. Write an if-statement that does the same as the following switch statement.

```java
switch (a) {
    case 1:
    case 3:
    case 7:
    case 17:
    case 79:
        trace("my favorite number");
        break;
    case 0:
    case 13:
    case 67:
    case 99:
        trace("not my favorite number");
        break;
    default:
        trace("just a number to me");
}
```

Q8. Write code to declare a 2d array to represent a tic-tac-toe game board, with all elements initialized to 0.

```java
// Example code
int[][] board = new int[3][3];
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        board[i][j] = 0;
    }
}
```

Q9. Write a switch statement that does the following:

```java
if rank is equal to 1 then
    trace out "A"
else if rank is equal to 2 to 10 then
    trace out the value of rank
else if rank is equal to 11 then
    trace out "J"
else if rank is equal to 12 then
    trace out "Q"
else if rank is equal to 13 then
    trace out "K"
else
    trace out "face down"
```
Q10. | What is the value of \( b \) when \( a = -9.4 \)? | What is the value of \( b \) when \( a = 9.4 \)? | What is the value of \( b \) when \( a = -9.6 \)? | What is the value of \( b \) when \( a = 9.6 \)? |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( b = \text{Math.abs}(a) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b = \text{Math.round}(a) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b = \text{Math.ceil}(a) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b = \text{Math.floor}(a) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b = \text{Math.sqrt}(a) )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Q11. The diagram below shows the inheritance relationships of three classes: \textit{PerformingArtist}, \textit{Dancer}, and \textit{Musician}.

![Inheritance diagram](image)

Given below are the \textit{incomplete} class definitions for \textit{Dancer} and \textit{Musician}.

\begin{itemize}
  \item \textit{Dancer}
  \item \textit{Musician}
\end{itemize}

(i) Complete the class definitions.

\begin{verbatim}
package {
  import flash.display.MovieClip;
  public class PerformingArtist extends MovieClip {
    private var payPerHour:Number;

    public function PerformingArtist(pay:Number) {
      payPerHour = pay;
    }
    public function perform():void {
      trace("perform");
      gotoAndStop("performing");
    }
  }
}
\end{verbatim}
package
{
    public class Musician
    {
        var instrument:String;
        function Musician(pay:Number, inst:String)
        {
            instrument = inst;
        }
        function perform():void
        {
            trace("play "+ instrument);
            super.perform();
        }
    }
}

package
{
    public class Dancer
    {
    }
}

(ii) Write code to instantiate each of the three classes in a Flash file—create one object from each class. Suppose the three instantiated objects are called aPerformingArtist, aMusician, aDancer, and each object has a different value of payPerHour.

(iii) What will be displayed in the Output panel when the following statements are executed?
aPerformingArtist.perform();
aMusician.perform();
aDancer.perform();

(iv) Will you get an error when the following trace() statement is executed. Explain why or why not.

trace(aMusician.payPerHour);

(v) Will you get an error when the following trace() statement is executed. Explain why or why not.

trace(aMusician.instrument);

(vi) Which statements below are valid?
aDancer.perform();
aPerformingArtist.perform();
Dancer.perform();
PerformingArtist.perform();
Q12.
package
{
  public class Kid
  {
    private var piggyBankFund:Number;
    private var collegeFund:Number;

    public function Kid()
    {
      piggyBankFund = 10;
      collegeFund = 100;
    }

    public function getAllowance(howmuch:Number):void
    {
    }

    public function sellLemonade(earning:Number):Void
    {
    }

    public function buyStuff(spending:Number):void
    {
    }

    public function checkPiggyBankBalance():Number
    {
      return piggyBankFund;
    }

    public function contributeToCollegeFund(howmuch:Number):void
    {
    }

    public function checkCollegeFundBalance():Number
    {
      return collegeFund;
    }
  }
}

(i) Write code to instantiate the Kid class to create two objects, myKid1 and myKid2.

(ii) Add a statement to each of the methods getAllowance(), sellLemonade(), buyStuff() to add to or subtract from the piggyBankFund property the amount that is passed into the method as parameter.
(iii) Add a statement to the method `contributeToCollegeFund()` to add to the `collegeFund` property the amount that is passed into the method as parameter.

(iv) Write a statement to invoke `myKid1.getAllowance()`. And the allowance amount is 30.

(v) Suppose after the above statement, there are trace statements like these:

```javascript
trace(myKid1.checkPiggyBankBalance());
trace(myKid2.checkPiggyBankBalance());
```

What will be displayed in the Output panel?

(vi) Write a statement to invoke `myKid1.contributeToCollegeFund()` adding the 10000 to `collegeFund`.

(vii) Suppose after the above statement, there is a trace statement like these:

```javascript
trace(myKid1.checkCollegeFundBalance());
trace(myKid2.checkCollegeFundBalance());
```

What will be displayed in the Output panel?

(viii) Add a keyword to the code to make the `collegeFund` property a static property.

(ix) With `collegeFund` being a static property, what will be displayed in the Output panel if you re-do (vi) and (vii) above?

Q13. Suppose in a card game, you have a deck of cards and you want to randomly draw 6 cards from the deck. Once a card is drawn, it is not available for the subsequent drawing.

The `cardDeck` array contains a list of the cards. `drawnCards` array will be used to store the 6 drawn cards.

Add code to the following to randomly select 6 cards from the `cardDeck` and store them in the `drawnCards` array.

```javascript
var cardDeck:Array = new Array("A", "K", "Q", "J", "ten", "nine",
"eight", "seven", "six", "five", "four", "three", "two");
var drawnCards:Array = new Array();
```

Q14. Suppose there is a movieclip instance, named `mc_kite`, on stage. Write code so that when you hold down UP or DOWN arrow with LEFT or RIGHT arrow keys, the kite will move diagonally accordingly.