

SAMPLE Final Exam
Computer Programming 230
Dr. St. John
Lehman College
City University of New York
Thursday, 17 December 2009

NAME (Printed) _____
NAME (Signed) _____
E-mail _____

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes.
- When taking the exam, you may have with you pens or pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- You may not use a computer or calculator.
- All books and bags must be left at the front of the classroom during this exam.
- **Do not open this exams until instructed to do so.**

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	
TOTAL	

1. True or False:

- (a) ___ In Alice, a loop never stops if the condition remains false..
- (b) ___ In Alice, an event can execute its statements only once.
- (c) ___ An array is the only data structure available in Alice.
- (d) ___ An object that responds when the event occurs is called a listener.
- (e) ___ In Java, any objects, including arrays, can be stored in an array.
- (f) ___ In Java, every class is automatically derived from the Object class.
- (g) ___ In Java, the parent class can use all methods defined in the child class.
- (h) ___ In Java, GUI components cannot contain other components.
- (i) ___ In Java, a try block in the try-catch statement can have only one catch block..
- (j) ___ In Java, an exception is thrown when a specified file is not found.

2. Line up the Alice statements with the corresponding statement in Java:

A

B

C

D

E

F

G

H

I `Mole moles[] = {mole, mole2, mole3, mole4, mole5, mole6, mole7, mole8}`

II `while (count < 10) {
 updateScore();
 count++;
}`

III `world.drawCircle();`

IV `moles[index].move(0.5,0.25);`

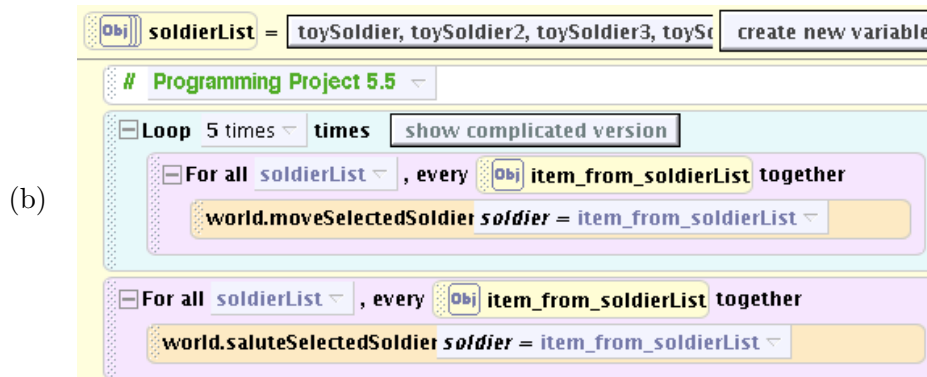
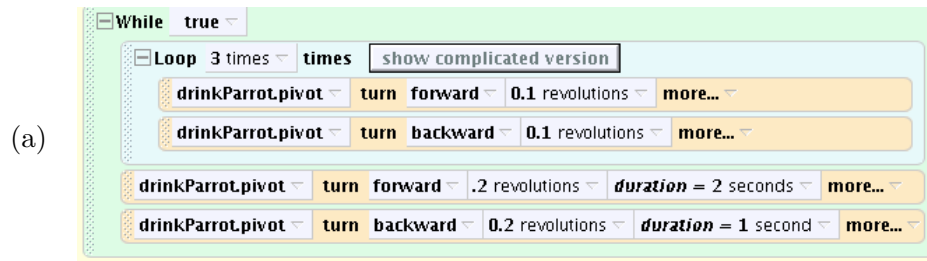
V `while (count < 10) {
 updateScore();
 count = count + 2;
}`

VI `x = Integer.toString(y);`

VII `x = x % 2;`

VIII `world.drawCircle(1,1);`

3. What happens when the code is run?



4. What is the output of the following code fragments:

(a)

```
int numtimes = 0;
while ( numtimes > 2 )
{
    System.out.print("Hi!");
    numtimes++;
}
System.out.print("Bye!");
```

Output:

(b)

```
boolean done = false;
int accum = 2;
while ( !done )
{
    if ( total > 20 )
    {
        done = true;
    }
    accum = 2*accum;
}
System.out.println(accum);
```

Output:

(c)

```
int i, j;
for ( i = 0 ; i < 4 ; i++)
{
    for ( j = 0 ; j < 4 ; j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Output:

(d)

```
int i, j;
for ( i = 0 ; i < 4 ; i++)
{
    for ( j = 0 ; j < 4 ; j++)
    {
        if ( (i+j)%2 == 0 )
        {
            System.out.print("*");
        }
        else
        {
            System.out.print("=");
        }
    }
    System.out.println();
}
```

Output:

5. What is the output?

```
(a) if ( ( 1 <= 1 ) && ( 0 > 10 ) )  
    System.out.println("Yes");  
    else  
        System.out.println("No");
```

Output:

```
(b) boolean tobe = true;  
    if ( tobe || !tobe )  
        System.out.println("Yes");  
    else  
        System.out.println("No");
```

Output:

```
(c) int x = 2, y = 3, z = 4;  
    if ( x+y*z > 15 )  
        System.out.println("Yes");  
    else  
        System.out.println("No");
```

Output:

```
(d) int number = 7;  
    boolean ispositive = ( number > 0 );  
    boolean iseven = ( number % 2 == 0 );  
    if ( ispositive && iseven )  
        System.out.println("Yes");  
    else  
        System.out.println("No");
```

Output:

```
(e) int year = 2000;  
    if (( year%4 == 0 && year%100 != 0 ) || ( year%400 == 0 ))  
        System.out.println("Yes");  
    else  
        System.out.println("No");
```

Output:

6. Assume the following class definition:

```
public class SampleClass {  
    public int number;  
    public String message;  
    public void print()  
    {    System.out.println(message + " " + number); }  
    public void mystery()  
    {    int i;  
        for ( i = 0 ; i < number ; i++ )  
            System.out.print(message);  
    }  
}
```

and the following code has been executed:

```
SampleClass first = new SampleClass();  
SampleClass second, third;  
first.number = 2;  
first.message = "What's up?";  
second = new SampleClass();  
second.number = 2*first.number;  
second.message = first.message.substring(0,4);  
third = first;
```

What is the output from the following statements?

(a) `first.print();`

Output:

(b) `first.mystery();`

Output:

(c) `second.print();`

Output:

(d) `second.mystery();`

Output:

(e) `third.print();`

Output:

7. (a) Write a **for**-loop that prints out the even numbers from 0 to 10:
0 2 4 6 8 10

- (b) Write a **while**-loop that reads in exam scores from the user and keeps a running total of the scores. The loop ends the user enters a negative number. After the loop, you should print the **average** exam score. Include declarations for all variables used.

8. You have just been accepted a job with the Procurement Unit of the United Nations (located in Copenhagen). Your first assignment is to keep track of automobile purchases. Your predecessor, before quitting, began writing an `Automobile` class. Each of the methods of the class is preceded by a comment that explains what the method should do. Fill in each method with the appropriate code:

```
public class Automobile
{
    public String maker; //The company who made the automobile
    public String model; //The model name (i.e. Explorer or Miata)
    public int year;     //The year the car was made
    public int numPass;  //Number of passengers car can legally carry
    public double price; //Price paid for car
    public double mpg;   //Miles per gallon (i.e. gas mileage) of car

    /* Prints all the information about the car: */
    public void print()
    {

    }

    /* Calculates the price per passenger */
    public double pricePerPass()
    {

    }

    /* Using the mpg, calculates the range (how far) given the amount of gas */
    public double range( double gallonsOfGas )
    {

    }

}
```


9. Create a new class called `Line` that extends the abstract class `Shape` below. Your `Line` class should have properties for the endpoints of a line, a constructor, and a method `draw()`.

```
//*****  
// Shape.java      Programming with Alice and Java  
//  
// Represents an abstract shape that can be drawn.  
//*****  
  
import java.awt.*;  
  
public abstract class Shape  
{  
    protected Color color;  
  
    //-----  
    // Descendents of this class implement the draw method to draw the  
    // particular shape in the specified graphics context.  
    //-----  
    public abstract void draw(Graphics gc);  
}
```

10. Write a **complete** Java program that asks the user for the name of a file and prints the contents of the file, **double spaced** to the screen: