Final Exam
COMP 14-062 Fall 2004
Dec 16, 2004

1. Closed book and closed notes.

2. Write all scratch work and answers on the exam itself. If you need extra space, let me know. Indicate your final answer by drawing a box or circle around it.

3. Write legibly. If I can't read it, you can't get credit for it.

4. Write your name on every page of this exam.

5. You do not need to put comments in any code you write. You will not be graded for coding style.

6. Assume any code segment is embedded in a correct program. That is, assume all variables have already been appropriately declared before they are used.

I pledge that I have neither received nor given unauthorized aid on this examination.

Signed: _____

Printed Name: _____

1) - Multiple Choice / Fill in the Blank [4 points each, 20 points total]

Circle the option that correctly fills the blank

1. A(n) _____ parameter defines what type must be passed to a method.

a) actual   b) formal   c) static   d) class

2. An easy way to share a variable between many instances of the same class is to declare the variable as _____.

a) public   b) private   c) static   d) instance data

3. A(n) _____ is a large, integrated collection of data.

a) network   b) graphics system   c) database   d) application

4. The presence of the _____ keyword indicates that it is okay to use a method or variable outside of the class in which it is defined.

a) static   b) final   c) public   d) private

5. Which of the following behaves most like an array when passed as a parameter?

a) int   b) String   c) char   d) static variable

2) Explain Concepts (17pts)
  a) [8 points]
Briefly explain the terms **class** and **object**, and describe how they're related to each other. You can use specific examples if you want.

  b)[9 points (3 each)]
Briefly describe the role of each of the following within a class:

   a) Instance variables

   b) Instance methods

   c) Constructors

3) [16 points (4 each)]
Write a method header (the very first line of a method) for methods that do each of the following things. Assume that all of these methods are not instance methods (that is, their headers start with the keywords public and static). Do not write the body of the method, just the header.

a.   A method named printX() that just displays the String "X" to the output window.

b. A method named doubleValue() that takes in an argument of type int and returns twice the argument's value.

c. A method named piCount() that takes in an array of doubles and returns the number of elements that are greater than Pi.

d. A method named largerThan() that takes in one int and one double and returns true if the int is larger than the double, and false otherwise.

4) Detect Errors: True or False (15pts)
If the code segment will compile and run with no errors, write true, otherwise write false. If the code segment will not compile, indicate why. (Each code segment is separate from the others)

	True or False
Example: int number = true; ← *different types*	_____false_____

1.)
```
double count = 0;
for( int i=0; i<10; i++ ){
    count += i / (i/2-1);
}
System.out.println( count );                    _____
```

2.)
```
char c = 'A';
int i = 10;
String str = "myString";
str += c + i;                                   _____
```

3.)
```
int sum = 0;
for( int i=0; i<100; i+=2 ){
    sum += i;
}
System.out.println( i );                        _____
```

4.)
```
int num = 10;
boolean valid = ( num < 5 ) && (num/2 == 2);

                                                _____
```

5.)
```
char ch = 'i';
if( ch >= 'a' && ch <= 'z' ){
    System.out.println( ch );
}                                               _____
```

5) [12 points] Evaluate the following expressions, given these variable initializations.

String str1 = "42, dude."
String str2 = "Bananas are for monkeys"
int num = 31;

a. int val1 =
     (int)((float)(num / 31 / 1 * 2 / 9) + (int)1.0);

b. String val2 = str2.substring(0, 1) +
  str2.substring(8, 12) +
  str2.substring(str2.indexOf("monkeys"));

c. boolean val3 = 30 / num < 7 == str1.length() >= 9;

d. String val4 = str2.substring(str2.indexOf("n"), 6);

6) [12 points]
Show the output produced by the following code.

```java
public class MyClass {

    public static void changeX() {
        int x = 20;
        System.out.println(x);
    }

    public static void incrementX(int x) {
        x++;
        System.out.println(x);
    }

    public static int returnX(int x) {
        x = 0;
        System.out.println(x);
        return x;
    }

    public static void main(String[] args) {
        int x = 10;
        changeX();
        System.out.println(x);
        incrementX(x);
        System.out.println(x);
        x = returnX(x);
        System.out.println(x);
    }
}
```

7) [15 points]
Suppose I've written a class named Adder. This class has three instance variables, all of type int, whose identifier names are num1, num2, and result. This class has a default constructor that initializes all of the class's instance variables to 0. This class has two instance methods: 1) go() takes two arguments of type int, adds the two arguments, and

stores their sum in the instance variable result; 2) display() prints the current value of result to the output window. Show the output produced by the following code. (Note: You can draw diagrams for partial credit, but the only thing that's required is the output itself.)

```
Adder add1 = new Adder();
Adder add2 = new Adder();
Adder add3 = add1;

add1.go(1, 2);
add3.display();
add2.num1 = 2;
add2.num2 = 4;
add1.go(add2.num1, add2.num2);
add1.display();
add3 = add2;
add2.go(add2.num1*2, add1.result);
add3.display();
add1.go(0, ++add1.result);
add1.display();
```

8) **[12 points]**
Write some code that will declare, initialize, and fill in an array of type int. After your code executes, the array should look as follows.

0	2	4	6	8	10	12	14	16	18

9) **[15 points]**
Show the output produced by the following code. Note: The second and third lines simply initialize arrays of length 5 containing the given values. Ask me if you don't understand those two lines.

```java
int i1 = 0, i2 = 0;
int[] array1 = {1, 2, 5, 5, 8};
int[] array2 = {2, 3, 4, 7, 10};
int[] array3 = new int[10];

for (int i3 = 0; i3 < 10; i3++) {
    if (i1 > 4)
        array3[i3] = array2[i2++];
    else if (i2 > 4)
        array3[i3] = array1[i1++];
    else {
        if (array1[i1] < array2[i2])
            array3[i3] = array1[i1++];
        else
            array3[i3] = array2[i2++];
    }
}

for (int i = 0; i < 10; i++)
    System.out.print(array3[i] + " ");
```

10) Show the output of the following code segment (20pts)

```java
import java.io.*;
class testCard
{
  public static void main(String[] args)
  {
    Card card1 = new Card(1, 1);
    int x = 2, y = 2;

    System.out.println("outside I: " + card1 + " x is " + x + " y is: " + y);
    update(card1, x, y);
    System.out.println("outside II: " + card1 + " x is " + x + " y is: " + y);
  }

  public static void update(Card card1, int  x, int y){
     System.out.println("inside I: " + card1 + " x is " + x + " y is: " + y);
     x = 3; y = 3;
     card1.setFace(x);
     card1.setSuit(y);
     System.out.println("inside II: " + card1 + " x is " + x + " y is: " + y);
  }
}
```

Given the card class:

```java
public class Card
{
   private int face;
   private int suit;

   public Card(int face, int suit)
   {
     // initialize the card
     this.face = face;
     this.suit = suit;
   }


   public String toString ()
   {
      // return a String that contains a text representation of the
card
     String cardName = "face: ";
     cardName += face + " suit: " + suit;
     return cardName;
   }

   public void setFace(int f)
   {
     face = f;
   }
```

```
      public void setSuit(int s)
      {
        suit = s;
      }

  }
```

11) **[10 points (5 each)]**

a.

Given the following array, how many times will values be swapped if the array is sorted using selection sort?

2	1	10	5	4	15

b)

How many times will a value be inserted into a new position if the same array is sorted using insertion sort?

12) [14 points] predict outputs

```java
public class SoyBean
{
    private static int numBeans = 0;
    private int myID;

    SoyBean()
    {
        myID = numBeans;
        numBeans++;
    }
    public int id() {return myID;}
}

public class Program
{
    public static void main(String[] args) {
        SoyBean b0 = new SoyBean();
        SoyBean b1 = b0;
        SoyBean b2 = new SoyBean();
        System.out.println("beans: " + b0.id() +
                           "," + b1.id() +","+ b2.id());
    }
}
```

13) [22 points]
Provide a definition for the method searchSorted, which takes as input (i.e., parameters) an array of integers *sorted in ascending order* and an integer target. The method returns true if there is at least one element of the array equal to the target. The method returns false otherwise. Use the fact that the array is sorted to possibly exit the search early. The searchSorted method should be both accessible outside of a class definition and available for invocation without using an object.

a)   write down the header of the method (7pt)

b)   write down your design from high level (8pt)

c) write down your code for the method (7pt)