

Final Exam
COMP 110-001 FALL 2006
Dec. 14, 2006

1. Closed book and closed notes
2. Write all scratch works and answers on the exam itself. If you need extra space, let me know. Indicate your final answer by drawing a box / circle around it to separate it from your scratch work.
3. Write legibly. If I can't read it, you will not get any credit for it.
4. You don't have to write comments in the codes.
- 5. It's always a good idea to show me the steps.**

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

Signed : _____

Printed Name : _____

1. Multiple Choice(20 points)

Identify the letter of the choice that best completes the statement or answers the question.

(1) Which of the following statements about a named constant is NOT true? _____

- a. Its content cannot change during program execution.
- b. Its value can be changed during program execution.
- c. It is a memory location.
- d. It is declared using the reserved word final.

(2) How many constructors can a class have? _____

- a. 0
- b. 1
- c. 2
- d. Any number

(3) How can a method send a primitive value back to the caller? _____

- a. It cannot send primitive values back to the caller
- b. By using the return statement
- c. By assigning the value to one of its parameters
- d. It can call its caller with the value

(4) In a Java program, the file name must be the same as the _____.

- a. method name
- b. class name
- c. object name
- d. data name

(5) What is the value of counter after the following statements executes? _____

```
counter = 0;
while (counter <= 50)
    counter = counter + 1;
```

- a. 48
- b. 50
- c. 51
- d. 53

(6) Which of the following is NOT true about return statements? _____

- a. A value-returning method returns its value via the return statement.
- b. Return statements can be used in void methods to return values.
- c. A method can have more than one return statement.
- d. Whenever a return statement executes in a method, the remaining statements are skipped and the method exits.

(7) Given the method heading

```
int larger(int x, int y)
```

which of the following does NOT demonstrate method overloading? _____

- a. int larger(int x, int y, int z)
- b. int larger(char x)
- c. int larger(int x, double y)
- d. double larger(int x, int y)

(8) Which of the following would be appropriate syntax for the heading of a copy constructor for a class called Ruler? _____

- a. public Ruler(int inches, int centimeters)
- b. public Ruler()
- c. public Ruler(Ruler myRuler)
- d. public copyRuler(int inches, int centimeters)

(9) Which of the following about Java arrays is true? _____

- (i) All components must be of the same type.
 - (ii) The array index and array element must be of the same type.
- a. Only (i)
 - b. Only (ii)
 - c. Both (i) and (ii)
 - d. None of these

(10) Consider the following declaration:

```
double[] sales = new double[50];  
int j;
```

Which of the following correctly initializes all the components of the array sales to 10. _____

(i)
for (j = 0; j < 49; j++)
sales[j] = 10;

(ii)
for (j = 1; j <= 50; j++)
sales[j] = 10;

- a. Only (i)
- b. Only (ii)
- c. Both (i) and (ii)
- d. None of these

2. Explain Concepts (8 points)

(1) Briefly explain the terms **class** and **object**, and describe how they're related to each other. You can use specific examples if you want.

(2) Describe the difference between shallow copying and deep copying.

3 . What is the output after executing following codes? (8 points)

(1)

```
int a = 2;
int b = 0;
int i = 0;
while (i < 5)
{
    System.out.println("a = " + a + " b = " + (++b));
    a = a + 2;
    i = a / 2;
    if (i > 6)
        continue;
    i = 2;
}
```

(2)

```
int x = 7;
boolean found = false;

do
{
    System.out.print(x + " ");
    if (x <= 2)
        found = true;
    else
        x = x - 5;
}
while (x > 0 && !found);
```

4. Searching (10 points)

Consider the following list

3 14 19 35 43 55 63 89 91 95 110

How many comparisons are required to determine whether the following items are in the list or not (both linear and binary search)? Assume that you do not know that the list is ordered for the linear search.

- | | | |
|-------|-----------------|-----------------|
| a. 19 | Linear search : | Binary search : |
| b. 45 | Linear search : | Binary search : |
| c. 63 | Linear search : | Binary search : |
| d. 3 | Linear search : | Binary search : |
| e. 95 | Linear search : | Binary search : |

5. Sorting (10 points)

Sort the following list using the bubble sort, the selection sort, and the insertion sort. Show the list after each iteration of the outer for loop.

28 18 5 42 1 30

- (a) Bubble sort
- (b) Selection sort
- (c) Insertion sort

6. Consider the following definitions of a recursive method. (8 points)

(a)

```
public static void mystery(int num)
{
    if (num > 0)
    {
        mystery(num - 1);
        System.out.print(num + " ");
    }
}
```

What is the output of the following statement?

```
mystery(4);
```

(b)

```
public static int mystery(int first, int last)
{
    if (first > last)
        return 0;
    else if (first == last)
        return first;
    else
        return first + mystery(first + 1, last - 1);
}
```

What is the output of the following statement?

```
System.out.println(mystery(6, 10));
```

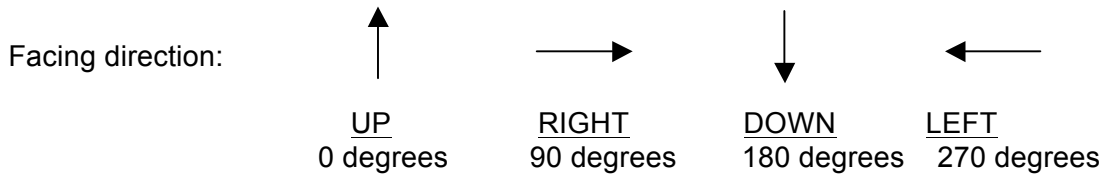
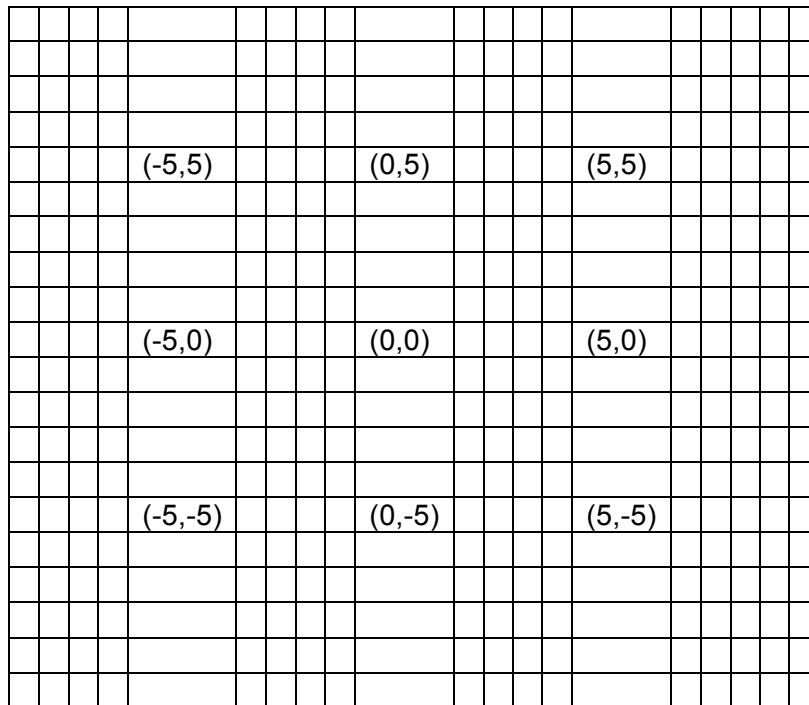

7. Writing a Class (18 points)

Imagine a rectangular grid like the one shown below. Each square has a unique row and column number (x,y). The square (0,0) is at the middle; x values increase to the right and decrease to the left; y values increase going up and decrease going down. A few (x,y) values are shown as examples.

We want to create a Robot class. Each Robot object represents one robot. Each robot has a location (x,y), and a direction in which it is facing (up, right, down, or left). A robot can move one square forward in the direction it is facing. It can also turn one quarter turn to the right. In addition, you can inquire about the robot's current x and y position and facing direction.

On the following pages, write the Robot class by filling in the data and method definitions. You need not do any error checking nor write comments.

***** If you use named constants for this class to make the code more efficient, I'll give you extra credit of 5 points *******



```
public class Robot
{
    // member variables
    // Current robot location and direction.

    // Constructors
    // No parameters: locate the robot at (0,0) facing up.
    public Robot()
    {

    }

    // Two parameters: locate the robot at (x,y) facing up.
    public Robot(int x, int y)
    {

    }

    // Methods
    // Move the robot one square in the direction it is facing.
    public void move()
    {

    }

    // Question continues on the next page.
```

```
// Continued from the previous page.

// Turn the robot one quarter turn to the right (e.g. from up
// to right; from right to down; from down to left,
// and from left to up).
public void turn()
{

}

// What is the robot's current x position?
public int whereX()
{

}

// What is the robot's current y position?
public int whereY()
{

}

// What direction is the robot currently facing
// (in degrees: 0, 90, 180, or 270)?
public int whatDirection()
{

}

} // End of Robot class
```

8. Using a class (12 points)

This question uses the Robot class, but does not require that you got the right answer to the previous question. It requires only that you understand the abstraction created by the Robot class.

- a. Write the code to create a Robot object r1 on grid square (0,0) facing up (0 degrees).

- b. Write the code to create a Robot object r2 on grid square (9,0) facing up (0 degrees).

- c. Write the code to move the Robots r1 and r2 toward each other so that they meet at squares (4,0) and (5,0). Leave the robots facing up.

- d. Write the code to create an array of Robots with 5 elements. Robots are created at (0,0), (1,0), (2,0), (3,0), (4,0).

- d. Which of the following should be defined static? (check all that applies)

- A method to move the robot backwards one square.
- A method to report how many Robot objects have been created so far.
- A toString() method to display the current robot position and direction in a nice form.
- A variable allowing the robot object to have a name (like "Robby" or "Tobor.")

9. (8 points)

A prime number (or prime integer, often simply called a "prime" for short) is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself. (More concisely, a prime number p is a positive integer having exactly one positive divisor other than 1). For example, the only divisors of 13 are 1 and 13, making 13 a prime number, while the number 24 has divisors 1, 2, 3, 4, 6, 8, 12, and 24 (corresponding to the factorization $24 = 2^3 \cdot 3$), making 24 *not* a prime number.

Write a method that uses a **loop** to decide whether a number is prime or not.

```
public static boolean IsPrime1(int num)
{
```

10. Recursion, decimal to binary (8 points)

In this program, we want to design and write a program that uses recursion to convert a non-negative integer in decimal format into the equivalent binary number. First we define some terms.

Let x be a non-negative integer. We call remainder of x after division by 2 the rightmost bit of x . Thus, the rightmost bit of 33 is 1 and rightmost bit of 28 is 0.

Following is an example of finding the binary representation of 35 (This is different algorithm from the one we learned in the class).

35 divided by 2 : quotient 17	remainder 1	↑	1 0 0 0 1 1
17 divided by 2 : quotient 8	remainder 1		
8 divided by 2 : quotient 4	remainder 0		
4 divided by 2 : quotient 2	remainder 0		
2 divided by 2 : quotient 1	remainder 0		
1 divided by 2 : quotient 0	remainder 1		

The rightmost bit of the first computation becomes the rightmost bit of the final answer. The second rightmost bit is computed from the quotient of the first computation and so on until the quotient becomes 0.

- (1) Fill in the blanks (underlined> in the following code that converts a non-negative decimal to the equivalent binary number (It is going to print the resulting binary number).

```
public static _____ decToBin(int num)
{
    if( _____ )
    {
        decToBin( _____ );
        System.out.print( _____ );
    }
}
```

- (2) What is the output of the following statement?

decToBin(69);

11. Method (10 points)

Write a method named `pickUnique` that takes an integer array and returns another integer array containing the input elements without any duplicates. For example, given `{10, 2, 7, 8, 2, 3, 7}` the method returns `{10, 2, 7, 8, 3}`.

Hint:

There could be two parts in this method.

- (a) In the first part, create an array that marks whether each element in the original array is unique or not. A double for-loop can be useful to compute the value of each element in this array.
- (b) In the next part, create an integer array that you will return. The size of this array can be computed while computing the array described above(a) . Value of each element is computed using the array above(a) and the original array given from the parameter.

