THE UNIVERSITY OF WESTERN ONTARIO
LONDON CANADA

COMPUTER SCIENCE 026a
FINAL EXAMINATION
DECEMBER 19, 2006
3 HOURS

NAME: ___Marking Scheme________________________________

STUDENT NUMBER: ____________________________________

Question

1-20. ________

21-40. ________

41. ________

42. ________

43. ________

44. ________

45. ________

46. ________

47. ________

48. ________

49. ________

TOTAL _________

(Out of 150 marks)

There are no cheat sheets, books, or other reference materials allowed for this exam. No calculators, cell phones, or other electronic devices are permitted either.

Part I – True/False – Determine whether the given statement is true or false. Circle your answer on this examination paper. [20 marks total, 1 mark each]

1. Every complete Java program must contain at least one `static` method.
   [✓ a. True.]
   b. False.

2. Wherever there is state equality between two object references, there must also be identity equality.
   a. True.
   [✓ b. False.]

3. The starting index position for an array in Java is 1.
   a. True.
   [✓ b. False.]

4. In the initialization portion of a `for` loop, it is possible to have multiple Java statements executed.
   [✓ a. True.]
   b. False.

5. In Java, `!(x < 20)` and `(x >= 20)` provide the same results.
   [✓ a. True.]
   b. False.

6. In Java, `!(!a || !b)` is equivalent to `a && b`.
   [✓ a. True.]
   b. False.

7. In Java, class names start with an uppercase letter, with the rest of the word being in lowercase. If there are multiple words, the first letter of each is also in uppercase.
   [✓ a. True.]
   b. False.

8. A method whose visibility is `public` can be referenced outside of the class in which it is defined.
   [✓ a. True.]
   b. False.

9. In defining a new class, you must explicitly define a constructor for that class.
   a. True.
   [✓ b. False.]

10. The `Object` class in Java has a `toString` method.
    [✓ a. True.]
    b. False.

11. If you do not specify the parent class of a new class that you define in Java, the new class will automatically be considered a subclass of the Java class `Object`.
    [✓ a. True.]
    b. False.

12. Constructors are special methods of a class that are automatically called when an object is created with the `new` operator.
    [✓ a. True.]
    b. False.

13. Accessor methods are public methods that return a data value.
    [✓ a. True.]
    b. False.

14. Mutator methods are private methods used in a class to change the value of a data field in that class.
    a. True.
    [✓ b. False.]

15. Text is stored in Java as a `String` object.
    [✓ a. True.]
    b. False.

16. The international standard for character representation used in Java is Unicode.
    [✓ a. True.]
    b. False.

17. Files are essentially named collections of bytes.
    [✓ a. True.]
    b. False.

18. Files are organized and grouped into directories.
    [✓ a. True.]
    b. False.

19. A path is the list of all of the directories from the root to the file.
    [✓ a. True.]
    b. False.

20. Exceptions are objects in Java.
    [✓ a. True.]
    b. False.

Part II – Multiple Choice – Choose the best answer from the choices given. Circle your answer on this examination paper. [20 marks total, 1 mark each]

21. Suppose that an entirely new kind of computer processor was created. Which languages would likely need to be introduced for this new processor?
    a. A new machine language
    b. A new assembly language
    c. A new high-level language
    d. All of the above
    [✓ e. Only a) and b)]

22. Which of the following is not a primitive data type in Java:
    a. `int`
    b. `double`
    c. `char`
    d. `boolean`
    [✓ e. All of the above are, in fact, primitive data types.]

23. Which of the following Java statements contains the invocation of a `static` method?
    a. `int minNumber = Math.min(3,4);`
    b. `System.out.println(name.toLowerCase());`
    c. `boolean result = Character.isLetter('A');`
    [✓ d. All of the above.]
    e. None of the above.

24. Suppose you had a while loop in a Java program that began with:

    ```
    while(true)
    ```

    This loop would:
    a. Never execute.
    b. Execute one time only.
    [✓ c. Likely loop indefinitely.]
    d. Generate an exception.
    e. None of the above.

25. When we use the term “nested loops”, we are referring to:
    a. Individual `for` or `while` loops.
    b. Multiple consecutive `for` or `while` loops.
    [✓ c. Multiple `for` or `while` loops, with the loops embedded inside each other.]
    d. All of the above.
    e. None of the above.

26. Which of the following practices contribute towards making what is considered to be a good method in Java?
    a. A method should do one and only one thing.
    b. A method should ideally be in the class that has the data the method is working on.
    c. A method should call other methods as necessary to do some of the work.
    [✓ d. All of the above.]
    e. None of the above.

27. Suppose we had the following method header:

```
public void someMethod(int someNumber)
```

Which of the following is an overloaded version of this method:
    [✓ a. `public void someMethod(int someNumber, int anotherNumber)`]
    b. `public int someMethod(int anotherNumber)`
    c. `public void someMethod(int anotherNumber)`
    d. `public void anotherMethod(int someNumber)`
    e. None of the above.

28. In Java, which of the following symbols is used as the logical negation operator:
    a. `?`
    [✓ b. `!`]
    c. `-`
    d. `^`
    e. None of the above.

29. Suppose we wanted to take a picture and make it look old and yellowed. To do this, what kind of image manipulation algorithm would we want to use?
    a. Posterize.
    b. Chromakey.
    c. Blur.
    [✓ d. Sepia-tone.]
    e. None of the above.

30. Suppose we wanted to cut out a small region of a picture to create a new picture. To do this, what kind of image manipulation algorithm would we want to use?
    [✓ a. Crop.]
    b. Rotate.
    c. Scale.
    d. Mirror.
    e. None of the above.

31. A constructor in Java:
    a. Is usually used to initialize the instance variables of an object.
    b. Is invoked with the help of the keyword `new`.
    c. Has the same name as its class.
    [✓ d. All of the above.]
    e. None of the above.

32. What does the Java statement:

```
Car rental = new Car();
```

most likely do?
    [✓ a. Declares a reference variable of type `Car` called `rental`.]
    b. Creates a new `Car` object without invoking a constructor in the process.
    c. Nothing; the statement is syntactically incorrect.
    d. Declares a reference variable of type `rental` called `Car`.
    e. None of the above.

33. A method that returns a value must contain the following keyword:
    a. `void`
    [✓ b. `return`]
    c. `break`
    d. `public`
    e. None of the above.

34. How many stars will the following loop display?

```
for (int star = 9; star < 0; star++) {
    System.out.println("*");
}
```

    a. 8
    b. 9
    c. 10
    [✓ d. 0]
    e. None of the above.

35. One wishes to output the contents of an integer variable if it has a value between 30 and 40, inclusive. Which of the following `if` conditions would accomplish this?
    [✓ a. `if ((number >= 30) && (number <= 40))`]
    b. `if ((number >= 30) || (number <= 40))`
    c. `if ((number <= 30) && (number >= 40))`
    d. `if ((number <= 30) || (number <= 40))`
    e. None of the above.

36. Suppose you had the following `for` loop in a Java program:

```
String message = "abcdefghijk";

for (int i=message.length()-1; i >= 0; i = i - ((i % 2)+1)) {
    System.out.print(message.substring(i,i+1));
}
System.out.println();
```

This loop would produce which of the following outputs:

a. `kjihgfedcba`

b. `kigeca`

c. `kjhgedba`

[✓ d. `kjhfdb`]

e. None of the above

37. Consider the following Java method:

```
public static boolean what(int[] arr) {
    for (int i = 0 ; i < arr.length ; i++)
    {
         if (arr[i] = 0)
         {
              return true;
         }
    }
    return false;
}
```

What does the method do?

a. It initializes each element of the array `arr` to `0`.

[✓ b. It returns `true` if at least one element of `arr` is `0`.]

c. It returns `true` if each element of `arr` is 0.

d. It always returns `false`.

e. None of the above.

38. The parameter to a main program is: `String[] args`. What is the type of `args`?

a. It is an array.

[✓ b. It is an array of strings.]

c. It is a string.

d. It is string of arrays.

e. None of the above.

39. What does the following code segment print, when executed?

```
String s = "Hello";
for (int i = s.length()-1 ; i >= 0; i--)
    System.out.print(s.substring(i));
```

a. `Helloellolloloo`
b. `Hello ello llo lo o`
c. `olleH`
d. `o l l e H`
[✓ e. None of the above.]

40. Consider the following Java program:

```
public class Findout {
     public static void main(String[] args) {

          String s = "Java is a lot of trouble";
          String rs = "";

          for (int i = s.length()-1; i >= 0; i--) {
               rs += s.charAt(i) ;
          }
     }
}
```

What does this code actually do?

a. It copies string `s` into `rs`.
b. Nothing, as this program does not compile correctly.
[✓ c. It reverses the order of the characters in `s` and puts the result in `rs`.]
d. It deletes the characters contained in `s`, and puts them in `rs`.
e. None of the above.

Part III – Short/Long Answer – Complete the following questions in the space provided on this examination paper.

41. Consider the following segment of Java code, taken from an instance method added to the Picture class. [10 marks total]

```
for (int x = 0; x < this.getWidth(); x++)
{
     for (int y = 0; y < this.getHeight(); y++)
     {
          Pixel pixel = this.getPixel(x,y);
          …
     }
}
```

a. Are the pixels in this picture accessed row-by-row or column-by-column? [1 mark]

Column-by-column

b. What are the (x,y) coordinates of the first three pixels accessed by this code? [3 marks]

(0,0), (0,1), (0,2)

c. What are the (x,y) coordinates of the last pixel accessed by this code? [1 mark]

(this.getWidth()-1, this.getHeight()-1)

d. Rewrite this code segment to access the pixels in this picture in the other order. That is, if they were accessed row-by-row above, provide code to have them accessed column-by-column, and if they were accessed column-by-column, provide code to have them accessed row-by-row instead. [5 marks]

```
for (int y = 0; y < this.getHeight(); y++)
{
  for (int x = 0; x < this.getWidth(); x++)
  {
       Pixel pixel = this.getPixel(x,y);
       …
  }
}
```

In essence, take off 1 mark for each mistake made in translating the code.

42. The following method Java method takes an array of integers as a parameter, and computes and returns the average of the integers in the array. This method is syntactically correct, but can generate at least three different run-time exceptions depending on how it is used. [10 marks]

```
public static int intArrayAverage(int [] intArray)
{
     int sum = 0;

     for (int i = 0; i <= intArray.length; i++)  (1)
     {
          sum = sum + intArray[i]; (2)
     }

     return sum/intArray.length;  (3)
}
```

a. Consider the following list of possible exceptions. Circle two exceptions that might be generated during the execution of this method and indicate where they could occur in the above method code. [4 marks]

[✓ ArithmeticException (3)]
[✓ ArrayIndexOutOfBoundsException (2)]
ClassNotFoundException
CommentTooLongException
DataFormatException
FontFormatException
IndexOutOfBoundsException
IOException
NegativeArraySizeException
NegativeZeroException
[✓ NullPointerException (1)]
StringIndexOutOfBoundsException
TimeoutException
UglyNameException
UndefinedVariableException
XPathException

Basically get 2 marks for each answer, 1 for exception and 1 for location

b. Rewrite the above method to avoid generating the exceptions identified and circled above. [6 marks]

```
public static int intArrayAverage(int [] intArray)
{
   int sum = 0;

   if ((intArray == null) || (intArray.length == 0))
        return 0;

   for (int i = 0; i < intArray.length; i++)
   {
        sum = sum + intArray[i];
   }

   return sum/intArray.length;
}
```

Basically get 3 marks for each correct correction made above..

43. Choose values for the integer variables `a, b, c` and `d` and the boolean variable `bb` to make the following expression evaluate to true. Use the table below to enter your values. [5 marks]

```
(!bb || (2*d < c && c-a>0)) && (((d-b != 0) && (a*b*2 > 100)) && bb)
```

Variable	Value
a	50
b	2
c	51
d	0
bb	true

44. Determining if a `Pixel` object is colored white is a relatively simple task: if a `Pixel` object's red, green, and blue components are all 255, then the `Pixel` object is white. [20 marks total]

a. Write an instance method for the `Picture` class called `countWhitePixels` that counts the number of white pixels in a picture. This method should take no parameters and will return an integer that is the number of white pixels in the picture. [10 marks]

```
public int countWhitePixels() {

  int count = 0;
  Pixel [] pixelArray = this.getPixels();
  Pixel pixelObj;
  for (int i = 0; i < pixelArray.length; i++) {
       pixelObj = pixelArray[i];
       if ((pixelObj.getRed() == 255) &&
           (pixelObj.getBlue() == 255) &&
           (pixelObj.getGreen() == 255)) {
            count++;
       }
  }

  return count;
}
```

b. Based on this result, write a new instance method for the `Picture` class called `countNonWhitePixels` that determines the number of pixels that are not white in a picture. This new method must call your new `countWhitePixels` method from above to determine how many pixels in the image are white, and then use this count in determining the number of pixels that are not white. This method should also take no parameters and will return an integer that is the number of pixels that are not white in the picture. [10 marks]

```
public int countNonWhitePixels() {

  int totalPixels;
  int whitePixels;

  totalPixels = this.getWidth()*this.getHeight();
  whitePixels = this.countWhitePixels();

  return totalPixels – whitePixels;
}
```

45. Consider the following Java code segment. [10 marks total]

```
for (int i = 0; i <= 10; i++)
{
     if ((i % 2) == 0)
     {
          System.out.println(i + " is even");
     }
}
```

a. What is the output from executing this code segment? [5 marks]

```
0 is even
2 is even
4 is even
6 is even
8 is even
10 is even
```

Basically deduct one mark for each mistake made.

b. Rewrite this code segment by changing the `for` loop to produce the same output, but without using an `if` statement. [5 marks]

```
for (int i = 0; i <= 10; i+=2)
{
     System.out.println(i + " is even");
}
```

Basically deduct one mark for each mistake made, or a couple for a big one.

46. Consider the following Java code segment. [10 marks total]

```
boolean continueLoop = true;
int count = 1;
while (continueLoop)
{
     System.out.println(count);
     count++;
     if (count > 10)
          continueLoop = false;
}
```

a. What is the output from executing this code segment? [5 marks]

```
1
2
3
4
5
6
7
8
9
10
```

Basically deduct one mark for each mistake made, or a couple for a big one.

b. Rewrite this code to eliminate the `continueLoop` variable and the `if` statement, while still using a `while` loop in the code. [5 marks]

```
int count = 1;
while (count <= 10)
{
     System.out.println(count);
     count++;
}
```

Again, basically deduct one mark for each mistake made, or a couple for a big one.

47. The following method added to the `Picture` class removes red-eye effects from a picture. It does so by examining the pixels in a region of a picture determined by the coordinates `(startX, startY)` and `(endX, endY)` inclusively. The method examines all the pixels in this region and replaces any that are close enough to `Color.red`, as determined by the method `pixelObj.colorDistance`, with the color of the parameter `newColor`. The program, however, has 4 logic errors. You should identify each and provide new code that would correct the errors. [10 marks]

```
public void removeRedEye(int startX, int startY, int endX,
                         int endY, Color newColor)
{
     Pixel pixelObj = null;

     // loop through the pixels in the rectangle defined by the
     // startX, startY, and endX and endY

     for (int x = startX; x <= endX+1; x++) {
          for (int y = startY; y <= endY+1; x++) {

               // get the current pixel
               pixelObj = this.getPixel(x,y);

               // if the color is near red then change it
               if (pixelObj.colorDistance(Color.red) < 167)
                    pixelObj.setColor(Color.red);
          }
    }
}
```

The four logic errors are:

- Having pixelObj.setColor(Color.red) … should be pixelObj.setColor(newColor)
- The second for loop should have y++ as its incrementing step and not x++
- Instead of endX+1 in the outer loop test, it should just be endX
- Instead of endY+1 in the inner loop test, it should just be endY

I guess we could go with 2.5 marks for each one … 1 mark for spotting the error, and 1.5 for correcting it properly.

48. Consider the following classes. [10 marks total]

```
Class AA {
     private int number;
     private String name;

     public AA () { // constructor for AA
      … … …
     }

     public int getNum () {
          return number;
     }

     public String getName() {
          return name;
     }
} // end of class AA

Class A3 extends AA {
     private int nsize;
     private String newName;

     public A3 () { // constructor for A3
     … … …
     }

     public String getName() {
          return newName;
     }

} // end of class A3

Class A4 extends AA {
     private boolean aFlag;

     public A4 () { // constructor for A4
     ... … …
     }

     public String toString () {
          return "A4 Object";
     }
} // end of class A4
```

Now, consider the following lines from a main program (numbered for reference), and answer the following questions. [Each question is worth 1 mark1]

```
1. AA x1 = new AA();
2. A3 x2 = new A3();
3. A4 x3 = new A4();
4. System.out.println(x1);
5. System.out.println(x2);
6. System.out.println(x3);
7. System.out.println(x1.getName());
8. System.out.println(x2.getName());
9. System.out.println(x3.getName());
```

a. In line 4, the `toString` method from which class is used to output `x1`? Object

b. In line 5, the `toString` method from which class is used to output `x2`? Object

c. In line 6, the `toString` method from which class is used to output `x3?` A4

d. In line 7, the `getName` method from which class is used to output the name of `x1?` AA

e. In line 8, the `getName` method from which class is used to output the name of `x2?` A3

f. In line 9, the `getName` method from which class is used to output the name of `x3?` `AA`

g. Give an example of an accessor method from one of the classes. getNum or getName

h. Based on the given class definitions, what are the instance variables associated with the object referenced by `x2?` `nsize, newName, number, name`

i. Based on the given class definitions, what are the instance variables associated with the object referenced by `x3?` `aFlag, number, name`

j. Write a mutator method for one of the given classes.

```
public void setNum(int n) { this.number = n; }
```

49. The following is the basic start of a class definition for describing a `Student`. Consider the class definition and then answer the following questions. [20 marks total]

```
Class Student {

      private final static MAX_COURSES;
      private int studentNumber;
      private String firstName;
      private String lastName;
      private int yearOfFirstEnrolment;
      private String faculty;
      private String [] courses = new String [MAX_COURSES];
      private int coursesCompleted;
      private double [] grades = new double [MAX_COURSES];

      public Student () { // default constructor for Student
         studentNumber = 0;
         firstName = "";
         lastName = "";
         yearOfFirstEnrolment = 0;
         faculty = "";
         courses = null;
         coursesCompleted = 0;
         grades = null;
      }

} // end of class Student
```

a. The following is the body of another constructor for class `Student`. Define the constructor's method header. [5 marks]

```
// another constructor for Student
// *** DEFINE THE HEADER THAT WOULD GO HERE


      public Student(int nVal, String fn, String ln)


      {
         studentNumber = nVal;
         firstName = fn;
         lastName = ln;
         yearOfFirstEnrolment = 0;
         faculty = "";
         courses = null;
         coursesCompleted = 0;
         grades = null;
      }
```

Get 1 mark for public, one for method name, and one for each parameter.

b. Consider the body of a method provided below and answer the following questions.
   i. What does the method do? [5 marks]

      This method counts the number of course grades in the grades array between the top and bottom marks inclusively.

   ii. Define the method's header. [5 marks]

      See below.

      1 mark for visibility, 1 for return type, 1 for name, and 1 for each parameter

   iii. Using the instance variables in the class definition, write an example of a use of this method. [5 marks]

```
public boolean readyToGraduate()
{
     if (this.countGrades(50, 100) >= 20)
         return true;
     else
         return false;
}
```

1 mark for calling method, one for each grade use, one for proper handling of return type and 1 more mark if no other errors

```
// a method in the class Student
// *** DEFINE THE HEADER THAT WOULD GO HERE


     public double countGrades(double bottom, double top)

     {
        if (this.coursesCompleted <= 0) {
           return 0;
        }
        else {
           double sum = 0;
          for (int nc=0; nc < this.coursesCompleted; nc++) {
             double g = this.grades[nc];
             if ((g >= bottom) && (g <= top)) {
                 sum++;
             }
          }
           return sum;
     }
```

This page has been left intentionally blank. Use it as additional workspace or extra space for answers if necessary.