

**Massachusetts Institute of Technology**

**African Internet Technology Initiative**

Summer 2007

# Final Exam

Wednesday, July 11<sup>th</sup>, 2007

Name: \_\_\_\_\_

Group: \_\_\_\_\_

You may use the class notes on this exam but you may **NOT** use your computer or any book. You have 2 hours to complete it. It contains 5 questions in 6 pages (including this one), totaling 100 points. Before you start, please check your copy to make sure it is complete. There are no compiler errors or runtime errors as all the code on the exam is legal Java code, unless otherwise noted.

**This exam will test your coding skills. You are expected to write legal Java code. Only provide what is asked of you. For some of the questions, you are not required to provide a class definition or method definition unless asked. Read carefully.**

**Write your name on the top of ALL pages.**

Please write neatly; we cannot give credit for what we cannot read.

*Good luck!*

Question	Total	Max
1		of 10
2		of 15
3		of 15
4		of 25
5		of 35
		of 100

1. (10 points) Write an `if` statement that will print “yes” to the screen if the `int` variable `x` is between 10 and 35 inclusive. If `x` is not between these values, then print “no”.

We will get you started by providing the beginning of the `if` statement. Do not write any code before the `if` statement. Assume that `x` is defined.

```
if (
```

2. (15 points) The method `Math.random()` returns a randomly distributed `double` between 0.0 (inclusive) and 1.0 (exclusive). We want to continuously call `Math.random()` until we find 10 values less than 0.4. Print how many trials (calls to `Math.random()`) were required to find these 10 values. Fill in the blanks in the code below for the implementation.

```
int trials = 0;
int under04 = 0;

while (under04 ____ ) {
    trials++;
    if ( _____ < 0.4 ) {
        _____;
    }
}

System.out.println("Trials:" + _____);
```

3. (15 points) Assume we have defined an array of `ints` named `values`. Write code that will sum the elements of `values` (beginning at index 0) **until** the running sum is greater than 100. Store the sum in the `int` variable `sum` (definition provided). Print the index which was the last to contribute to the sum. It follows that if the sum of the entire array is less than 100, then the code will print the size of the array minus one (the last index of the array).

```
int sum = 0;
```

4. (25 points) Write a method named `moreThanOnce` that takes an array of `ints` named `values` and an `int` named `key` as arguments. The method determines if `key` appears more than once in the `values` array. If it does, return `true`, otherwise return `false`. We will provide you with some of the definition for the method. Fill in the blanks and provide the implementation of the method.

```
public _____ moreThanOnce( _____ values,  
                               int key)
```

```
{
```

```
}
```

5. (35 points) Write a class that represents a rectangle with length and width. Include the following:
- Fields for the length and width as `doubles` that are accessible from any class
  - A constructor with two double arguments that will set the initial values of the length and width fields
  - A method that returns the area of the rectangle: `int area()`
  - A method that multiplies the length and width by an `int` argument:  
`void scale(int factor)`

All methods should be accessible from any other class. We will start the class for you. Complete the implementation of the class.

```
public class Rectangle {
```

```
}
```