Worksheet problem:

A very long simulation took 1902073 seconds. Write a program to figure out how many days, minutes and seconds it took, and output the answer. Use the operators "/" and "%" to do this.

If you don’t know how to start, try a simpler example first.

How many minutes and seconds is 70 seconds? How did you get this?

\[
70 \text{ seconds} / 60 \text{ seconds per minute: } 70/60 = 1
\]

\[
70 - 60 = 10 \text{ seconds left over}
\]

(see Work1.java)
Reading input from Console:

```
Scanner input = new Scanner(System.in);
System.out.print("Enter a double value: ");
Scanner input = new Scanner(System.in);
double d = input.nextDouble();
int i = input.nextInt();
String line = input.nextLine();
```
Other statements and operators (needed for HW2):

• boolean types: true or false;

```java
boolean foundBigNum = false;
if (num < 100) foundBigNum = true;
```

boolean expression using relational operators: > < >= <= == !=

no need for `if (found == true)`
instead use `if (found)`

• if statements (with or w/o else clause)

indent inside if/else statement
curly braces - can help prevent mistakes

• variations of if statements

```java
if (num < 100) {
    if (found) {
        foundBigNum = true;
        prevNum = num;
    }
    else {
        foundBigNum = false;
    }
}
```
Other statements and operators (cont.):

- nested if statements  (several styles possible)

```java
if (grade >= 90)
    System.out.println(" grade is A");
else
    if (grade >= 80)
        System.out.println(" grade is B");
    else
        if (grade >= 70)
            System.out.println(" grade is C");
        else
            if (grade >= 60)
                System.out.println(" grade is D");
            else
                System.out.println("grade if F");
```

```java
if (grade >= 90)
    System.out.println(" grade is A");
else if (grade >= 80)
    System.out.println(" grade is B");
else if (grade >= 70)
    System.out.println(" grade is C");
else if (grade >= 60)
    System.out.println(" grade is D");
else
    System.out.println("grade if F");
```
Other statements and operators (needed for HW2) (cont.):

• dangling else problem

if (b1)
if (b2)
System.out.println("A");
else
System.out.println("B");

• if b1 and b2 both true, A is output
• what if not both true?

Do not do this in practice!

Use proper indentation.
Other statements and operators (needed for HW2) (cont.):

- dangling else problem

```java
if (b1)
if (b2)
System.out.println("A");
else
System.out.println("B");
```

- if b1 and b2 both true, A is output
- what if not both true?

- Rule is that else clause goes with more recent unclosed if in same block

```java
if (b1)
    if (b2)
        System.out.println("A");
    else
        System.out.println("B");
```
Other statements and operators (needed for HW2) (cont.):

• dangling else problem

if (b1)
if (b2)
System.out.println(“A”);
else
System.out.println(“B”);

• if b1 and b2 both true, A is output
• what if not both true?

• Rule is that else clause goes with more recent unclosed if in same block

• For else with first if need curlies (and fix indentation)
Math.random() examples:

Random number generator is only producing pseudo-random numbers.

\[ d = \text{Math.random()} \] returns a double \( d \), where \( 0 \leq d < 1.0 \)

• What if you want a double between 0 and 9 (exclusive)?

• What if you want a double between 5 and 14 (exclusive)?

• What if you want an integer between 5 and 14 (exclusive)?
Math.random() examples:

Random number generator is only producing pseudo-random numbers.

\[ d = \text{Math.random();} \quad \text{returns a double } d, \quad \text{where } 0 \leq d < 1.0 \]

- What if you want a double between 0 and 9 (exclusive)?
  \[ d = 9 \times \text{Math.random();} \]

- What if you want a double between 5 and 14 (exclusive)?
  \[ d = 9 \times \text{Math.random();} + 5; \]

- What if you want an integer between 5 and 14 (exclusive)?
  \[ d = \text{(int)} (9 \times \text{Math.random();} + 5); \]

(see Dice.java example)
Math.random() examples:

• This also introduces the while loop:

```java
import java.util.Scanner;

public class Dice {

    public static void main (String[] args){

        boolean won = false;
        Scanner keyInput = new Scanner(System.in);
        int guess, dice;

        while (!won){ // boolean initialized to false so play at least once

            // get the guess
            System.out.println("Enter guess for throw of dice: ");
            guess = keyInput.nextInt();

            // throw the dice
            dice = (int) (Math.random() * 6 + 1);

            // do we have a winner?
            if (guess == dice){
                System.out.println("You won!");
                won = true;
            }
            else {
                System.out.println("You lost :(");
            }

        }/* end while */
    }/* end main */
}/* end class */
```
for loops:

for (i=initialValue; i < endValue; i++){
    // loop body
}

• If i not previously declared, and not used elsewhere can say
  int i = initialValue;

• other possibilities: i+=2; do more than one thing, e.g.  i++, j++

• do not modify loop index inside loop

Why use one over the other?

equivalent while loop

i=initialValue;

while (i < endValue){
    i++;
    // loop body
}

} /* end while */