# Chapter 3: Using Classes and Objects .
# Lab Exercises

1. In many situations a program needs to generate a random number in a certain range. The Java Random class lets the programmer create objects of type Random and use them to generate a stream of random numbers (one at a time). The following declares the variable *generator* to be an object of type Random and instantiates it with the *new* operator.

```
Random generator = new Random();
```

The *generator* object can be used to generate either integer or floating point random numbers using either the *nextInt* method (either with no parameter or with a single integer parameter) or *nextFloat* (or *nextDouble)* methods, respectively. The integer returned by *nextIn(t* could be any valid integer (positive or negative) whereas the number returned by *nextInt(n)* is a random integer in the range 0 to n-1. The numbers returned by *nextFloat()* or *nextDouble()* are floating point numbers between 0 and 1 (up to but not including the 1). Most often the goal of a program is to generate a random integer in some particular range, say 30 to 99 (inclusive). There are several ways to do this:

- **Using nextInt():** This way we must use the % operator to reduce the range of values—for example,

  ```
  Math.abs(generator.nextInt()) % 70
  ```

  will return numbers between 0 and 69 (because those are the only possible remainders when an integer is divided by 70 - note that the absolute value of the integer is first taken using the *abs* method from the Math class). In general, using % N will give numbers in the range 0 to N - 1. Next the numbers must be shifted to the desired range by adding the appropriate number. So, the expression

  ```
  Math.abs(generator.nextInt()) % 70 + 30
  ```

  will generate numbers between 30 and 99.

- **Using nextInt(70):** The expression

  ```
  generator.nextInt(70)
  ```

  will return numbers between 0 and 69 (inclusive). Next the numbers must be shifted to the desired range by adding the appropriate number. So, the expression

  ```
  generator.nextInt(70) + 30
  ```

  will generate numbers between 30 and 99.

# Exercises

## FunnyStrings

Write a program called “FunnyString” that asks the user to enter a favourite colour, food, animal and the first name of a friend. Use a Scanner object and read one thing per line.
The program should than print the following:
For example if the user entered **blue** for colour, **hamburger** for food, **dog** for the animal and **Jake** for the fried the output will be:

*I had a dream that Jake ate a blue dog*
*and said it tasted like hamburger !*

## Email

Write a program that reads a String from the input containing a name, and compute the email address with the first 3 letters of the first name the 4 letters of the last name and of course the domain.
If I run the program with my name Nicolina Mansson as input I will get the following email address
**Nic-Mans@hotmail.com**

**Note:** Use Scanner object to read input

Use methods from class String : **indexof()** to delimitate the name , **substring()** to get the letters in the first name and last name , **concat()** to put together the emai address ( you can concatenate by using + as well) .

## PigLatin , your first crypto algorithm

Write a program that starts with the sting variable *first* set to your first name and the string variable *last* set to your last name. Both names should be at lowercase. Your programme should than create a new string that contains your full name but in **pig latin**. The first letter both for first and last name should be capitalized.
To convert to **pig latin** you move the first letter to the end of the word and then ad “ay”.
Use the substring () and toUpperCase() methods to construct the new name.
Output the new name to the screen. For example if:
first = ” walt”
last =”savitch”

the programme should output “Altway Avitchsay”

Change the programme so programme will be more general means that the user will input his own name and get it in ***pig latin.***

## SumOfDigits

Write a program that reads a four-digit number (such 1998) as a String and than display it one digit per line like so:

1
9
9
8

To do that *use charAt() metod. So if you reed the number into the String variable called for example **textNumber** you should call the method you will get the first digit by using*

***char digit1=textNumber.charAt(0);***

b) A character, the data type **char**, ‘a’, ‘r’, ect, is represented internally as codes (sequences of 1 and 0) witch in the end is a numerical value. For example the char ‘a’ has the numerical value 97 and ‘F’ has 70.
That was not the case in the early days of computing, different computers used different coding schemes and character sets.
To avoid this problem , U.S. standardized some coding schemes, one of this is ASCII ( American Standard Code Information Interchange). The ASCII table contain 128 both printable and not printable character. To accommodate the character symbols of non-English languages, a new standard established, the Unicode Worldwide Character Standard.

So, now print the ascii code of your digit? **Confused?** To print the character ‘a’ I will write in my programme ***System.out.println((int) a);***

So, if you want to print the Unicode value of the digit1 you should use

***System.out.println((int) digit1);***

c. Now count and print the sum of the digits. To do that you need to convert the digit ( of type char ) to a really numerical value ( of type int)

int tal1 = Integer.parseInt( digit1+””);

*We use Integer.parseInt() to convert Strings to int.*
*We use digit1+”” , to convert a char to a String*

# RightTriangle

```
   // *******************************************************************
   //   RightTriangle.java
   //
   //   Compute the length of the hypotenuse of a right triangle
   //   given the lengths of the sides
   // *******************************************************************
   import java.util.Scanner;

   public class RightTriangle
   {
      public static void main (String[] args)
      {
         double side1, side2;  // lengths of the sides of a right triangle
         double hypotenuse;    // length of the hypotenuse

              Scanner scan = new Scanner(System.in);

         // Get the lengths of the sides as input
         System.out.print ("Please enter the lengths of the two sides");
         side1=
         side2=
        // Compute the length of the hypotenuse.
        //Use  Math.sqrt( … ) för beräkning av kvadratroten.

     hypotenuse=
        // Print the result
         System.out.println ("Length of the hypotenuse: " + hypotenuse);
       }
}
```

In many situations a program needs to generate a random number in a certain range. The Java Random class lets the programmer create objects of type Random and use them to generate a stream of random numbers (one at a time). The following declares the variable *generator* to be an object of type Random and instantiates it with the *new* operator.

```
      Random generator = new Random();
```

**Now:** Change the program so you create a **Random-object i**nstead of the **Scaner- object** as show it before.

Use the **generator.nextInt(** 20**)** to generate random numbers for the side1 and side2 variables than compute hypotenuse as before.

# Rolling Dice

Write a complete Java program that simulates the rolling of a pair of dice. For each die in the pair, the program should generate a random number between 1 and 6 (inclusive). It should print out the result of the roll for each die and the total roll (the sum of the two dice), all appropriately labeled. You must use the Random class.

The Random class is a part of the **java.util** package. Don't forget to **import** it. Use the Random class, nextInt(…) method .

## Invest

**If** you invest P dollars at R percent interest rate compounded annually, in N years, your investment will grow to **P (1+(R/100.0)) ^N** dollars . Write a application that accepts P, R and N ( P as double and R,N as integers) and compute the amount of money earned after N years. Use a scanner object för input. Use **Math.pow()** to compute power of. Use **DecimalFormat** to print the result with 2 decimal digits.

## Date ( om du hinner med, inte obligatoriskt)

Write a program that reds a string for a date in the format month/day/year and displays it as in the format day.month.year, which is a typical format used in Europe.
For example if the input is 06/17/08 the output should be 17.06.08 .

*Hint: Search in the class String for the appropriate method to solve this problem.*