

# JAVA PROGRAMMING

FOR

# ICS3M

**COMPUTER AND INFORMATION SCIENCE**

PRESENTATION



John Ketelaars  
Consultant  
Computer Science

## Acknowledgments and Preface

These pages are excerpted (and slightly changed) from a course manual written for the Toronto District School Board.

I have followed the *Ministry sponsored Course Profile* as to Units and followed their organization of expectations.

Java is receiving a lot of attention worldwide, and many teachers are now becoming more and more familiar with it.

In the examples listed, I have avoided implementing *object oriented techniques*, leaving that to grade 12. This has made certain solutions to problems a bit awkward, especially in units 4 and 5, where files should be more naturally turned into objects.

Java does not have its own console (for the default input and output) and therefore Sun Systems, the creators of Java, recommend third party integrated software: for the Macintosh: Metrowerks' Codewarrior and for the IBM compatibles: Codewarrior, IBM's VisualAge Java, and others.

I recommend Holt Software's "**Ready to Program with Java**" (it is available separate from the book) It is fast, and runs very well on old machines with minimal memory.

John Ketelaars  
Whitby, Ont.  
July 2001

<b>Unit 2: Beginning to Program</b>	<b>1</b>
<b>Unit 3: Problem Solving with Procedures and Functions</b>	<b>12</b>
<b>Unit 4: Information Storage and Related Issues</b>	<b>20</b>
<b>Unit 5: Using Data Structures</b>	<b>22</b>
<b>Unit 6: Putting It All Together</b>	<b>26</b>
<b>Appendix: zodiac.txt</b>	<b>29</b>

# Unit 2: Beginning to Program

Time: 24 days

## Unit Description

This unit focuses on basic programming structures. Students write simple programs, using variable assignment, repetition, and decision structures, and develop effective testing, validating, and documenting skills. They also explore roles of effective communicators and reflective thinkers when following a problem-solving model (e.g., user inputs a series of marks, each value is validated, the average is calculated, and a grade is assigned).

### 2.1 Input and Data Storage in Simple Programs

6 days

**Teaching points:** System.out.print, short, int, long, float, double, char, String, Input  
Introduce internal documentation.  
Introduce casting

2.1.1 Show the Standard "Hello World" program, and make the point, that many features will stay the same for the next set of programs.

#### (Output)

2.1.2 Write a program that uses stars (\*) to print your initials; a diamond, a triangle or any other interesting shape

#### (Arithmetic)

2.1.3 Write a program that assigns integer variables a,b values and print the sum, difference, product, and modulus

#### (Input)

2.1.4 Write a program that lets you enter the radius of a circle and that prints both the circumference and the area of the circle

#### (Careful how you mix integers with reals: $5/9 = 0$ )

2.1.5 Write a program that lets you enter a degree measure in Fahrenheit, and that prints the result in celsius degree measure:  $c = \frac{5}{9}(f - 32)$

2.1.6 Write a program that lets you enter a distance in kilometers and that converts it into miles.

#### (Mod)

2.1.7 Write a program that lets you enter a number of hours, and that converts it to days and hours. For example, 111 hours = 4 days and 15 hours.

2.1.8 Write a program that lets you enter a number of minutes, and that will calculate the number of days, hours and minutes that represents.

**(Math functions)**

- 2.1.9 Write a program that lets you enter the two sides of a right angled triangle, and that prints the hypotenuse.
- 2.1.10 Write a program that lets you enter the three sides of any triangle, and that prints the area of the triangle using Heron's formula:  $\sqrt{s(s-a)(s-b)(s-c)}$ , where  $s$  stands for half the perimeter:  $\frac{a+b+c}{2}$
- 2.1.11 Write a program that converts degree measures to radian measures
- 2.1.12 Write a program that uses the `Math.sin()` function to calculate the area of a triangle, given two sides and the contained angle: If the sides are  $a, b$  and the angle is  $C$ ,  
then the area is:  $\frac{ab \sin C}{2}$
- (use `Math.cos()`)**
- 2.1.13 Write a program that lets you enter two sides of a triangle and the contained angle, and that will use the cosine law to calculate the third side: If the sides are  $a, b$  and the angle is  $C$ , then the third side is  $\sqrt{a^2 + b^2 - 2ab \cos C}$
- (use `Math.pow()`)**
- 2.1.14 Write a program that let you enter two numbers  $a, b$ , and that prints out  $a^b$
- 2.1.15 Write a program that lets you enter a number and that rounds it off to 3 digits behind the decimal point.

## 2.2 Introducing Repetition

6 days

### Teaching points:

Introduce the **for( ; ; )** loop; introduce `++` , `++l`, `i--`, `--i`, `a+=b`, `a-=b`, `a*=b`, `a/=b`, `a%=b`; introduce Strings with concatenation, the `length()` method and the `substring()` method. Explain that Strings are objects, and methods do things to its objects, in this case, reveal something about them.

2.2.1 Write a program that prints the numbers from 1 to 10

2.2.2 Write a program that prints the numbers from 10 to 1

2.2.3 Write a program that prints the numbers from 1 to 30 by 2's, by 3's etc.

2.2.4 Write a program that prints a conversion table for Fahrenheit to Celsius

2.2.5 Write a program that prints a conversion table for Celsius to Fahrenheit

2.2.6 Write a program that prints the sum of the 5 numbers that you entered:

2.2.7 Write a program that prints the average of the 5 numbers that you entered

(use **Math.max()**)

2.2.8 Write a program that prints the largest of the 5 numbers that you entered

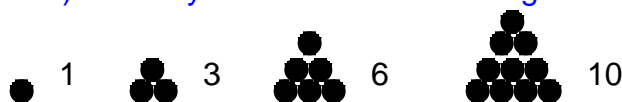
(use **Math.min()**)

2.2.9 Write a program that prints the smallest of the 5 numbers that you entered

2.2.10 Write a program that lets you enter a positive number  $n$  and that prints out  $n!$  ( $1!=1$ ,  $2!=1 \times 2=2$ ,  $3! = 1 \times 2 \times 3=6$ ,  $4!=1 \times 2 \times 3 \times 4=24$ ,  $5! = 1 \times 2 \times 3 \times 4 \times 5=120$ , etc.)

2.2.11 Write a program that lets you enter a positive number  $n$  and prints the  $n$ th triangular number:

(the 3rd triangular number has a base of 3, the  $n$ th triangular number has a base of  $n$ ). Hint - you must add something to the last one to get the next one.



2.2.12 Write a program that lets you enter a number  $n$  and that calculates the  $n$ th Fibonacci number: 1,1,2,3,5,8,... Each new number is the sum of the last two.



## 2.3 Introducing Selection

2 days

### Teaching points:

- Introduce if statement
- compare integers, reals and chars
- introduce `&&` and `||`
- introduce the else statement
- emphasize the difference between single quotes and double quotes
- introduce the string methods: `charAt()`

- 2.3.1 Write a program that lets you enter a mark between 0 and 100 and that will test its value and print appropriately: "PASS" or "FAIL"
- 2.3.2 Write a program that will let you enter any number and that will print its square root or the message: "Negative numbers do not have a square root value"
- 2.3.3 Write a program that will let you enter 3 sides of a triangle. if each of the three sides is less than half the perimeter, then a triangle can be formed (2,3, 1000 can not form a triangle) If a triangle can be formed, find the area of the triangle. If no triangle can be formed, say so. (see problem 2.1.10 )
- 2.3.4 Write a program that lets you enter 5 marks between 0 and 100 and that prints out in each case, the letter grade:  
e.g. F = 0-49, D = 50-59, C = 60-69, B = 70-79, A = 80-89, A+ = 90-100
- 2.3.5 Write a program that finds the roots of the equation  $Ax + Bx + C = 0$  by entering the values of A, B, C  
if  $B^2 - 4AC = 0$  then there is only one root, which is:  $-\frac{B}{2A}$   
if  $B^2 - 4AC > 0$  then there are two roots:  $\frac{-B + \sqrt{B^2 - 4AC}}{2A}$  and  $\frac{-B - \sqrt{B^2 - 4AC}}{2A}$
- (charAt())**
- 2.3.6 The students of your class are to be subdivided into 3 teams according to their first name. Names starting with A-I belong to team 1, Names starting with J-P belong to team 2 and Names starting with Q-Z belong to team 3.  
Write a program that lets you enter 5 names such as: Moe and that will print out for example: Moe belongs to team 2
- 2.3.7 Write a program that lets you enter five words and that will print in each case that word, if it contains more or less than 4 letters. If it contains four letters, it will print: XXXX

## 2.4 Combining Selection and Repetition

6 days

### Teaching points:

introduce the function **Math.random()**

While loop

Do-while loop

introduce the String functions **equals()**, **equalsIgnoreCase()** **compareTo()**

introduce arrays

- 2.4.1 Write a program that lets you print 20 random real numbers between 0 and 100
- 2.4.2 Write a program that lets you print 20 random integers between 1 and 100 inclusive.
- 2.4.3 Write a program that lets you generate 100 pairs of dice  
If the pair adds up to 2, print "snake eyes!"  
If the pair adds up to 7, print "lucky seven"  
if the pair adds up to any other sum, do not print out anything
- 2.4.4 Write a program that lets the computer generate a secret number (an integer between 1 and 100, for your friend to guess. If they guess too high, print: "too high, guess again." If they guess too low, print: "too low, guess again" If they guess the number, let the program stop and print "congratulations"
- 2.4.5 Modify the program so that it will count and print out the number of guesses it took to find the random number
- 2.4.6 A certain variation of the game of 21 is played as follows The computer generates two random numbers between 2 and 11. If the sum is between 18 and 21 then you win. If the sum is less than 18 you may ask for either 1 or 2 additional numbers by typing "1" or "2". If the sum is still under 18 then you lose. If the sum is over 21 then you win. Write a program to simulate this game.
- 2.4.7 In a variation of the game of NIM, two players take away alternatively 1,2 or 3 coins from a given pile of pennies. The person making the last move wins. Write a program to simulate this game, starting with a random number between 10 and 20.
- 2.4.8 Write a program that lets you enter an integer and that prints all the divisors of that number. Note that  $d$  is a divisor of  $n$  if  $n \% d = 0$

- 2.4.9 Write a program that lets you enter two integers and that prints the greatest common divisor of these integers. Note that the greatest common divisor of two numbers is the second last number in the sequence of numbers defined as follows: the first number is the largest of the two, the next number is the smallest of the two, each succeeding number is the remainder when the second last number is divided by the last number. The sequence ends with 0. It is called the **Euclidian algorithm**.  
Example: The GCD of 45 and 55 is 5, from the sequence: 55, 45, 10, 5, 0  
The GCD of 222 and 90 is 6 from the sequence: 222, 90, 42, 6, 0
- 2.4.10 Write a program that lets you enter two integers,  $a, b$  representing the fraction  $\frac{a}{b}$  and that uses the Euclidian algorithm (see 2.4.9) to reduce the fraction
- 2.4.11 Write a program that lets you enter an integer, and that will print appropriately "is a prime number" or "is a composite number". A number  $n$  is a prime number, if it has no divisors  $d$ , where  $d \leq \sqrt{n}$ , or more conveniently, where  $d^2 \leq n$
- 2.4.12 Write a program that lets you enter a "login" and "password". The Login ignores upper or lower case, but the password must be exact. After entering both, your program must say either; "**welcome**" or "**password or login incorrect**"
- 2.4.13 Write a program that lets you enter a word and that checks if is a palindrome (the same forward as backwards, such as Mom, Madam, Radar). The program should repeat asking for more words, until you enter "stop". The test should be case insensitive.
- 2.4.14 Write a program that lets you enter a word and that prints out the number of vowels and the number of consonants (vowels are: a,e,i,o,u. all others are consonants)  
The program should repeat asking for more words, until you enter "stop"
- 2.4.15 Write a program that lets you enter an integer and that will translate it to its spoken form from right to left. Example:  
Enter a number: 32457  
32457 = seven five four two three
- 2.4.16 Modify the program in 2.4.15 so that the words occur in the proper order:  
Enter a number: 32457  
32457 = three two four five seven

- 2.4.17 A deck of cards contains 4 suits (clubs, diamonds, hearts, spades) of 13 cards each (2,3,4,5,6,7,8,9,10,Jack, queen, king, ace). The lowest value card is the 2 of clubs, followed by the 3 of clubs and so on. The 14th card is the 2 of diamonds, the 28th card is the 3 of hearts and the highest card is card #52, the ace of Spades.  
Write a program to print out a random set of 5 cards. (challenge: make sure the cards are all different)
- 2.4.18 Write a program that will let your computer randomly construct sentences such as: The cow jumped over the moon, by selecting the nouns, verb and preposition randomly from arrays. Create three arrays, containing 10 alternatives respectively for (1) nouns such as: cow and moon, (2) verbs, such as: jumped, and (3) prepositions: over, under, around, below, by, with, to, in, into, outside, inside  
Your program should print out sentences, in a loop and ask to continue until you say "n"  
the moon smiled under the horse  
continue? (y/n) y  
the pig drove in the car  
continue? (y/n) n  
the end!
- 2.4.19 In statistics, the standard deviation is a measurement of spread. If in a class students have an average of 70% on a test, and the standard deviation is 15, then that indicates that students who got more than 85 and students who got less than 55 are somehow exceptional. The standard deviation is calculated as follows:  
Assume the numbers are  $x[0], x[1], x[2], \dots$   
First find the average  $m$  of the set of numbers. Then find the square of the differences of all the numbers from  $m$  i.e.  $(x[0] - m)^2, (x[1] - m)^2, (x[2] - m)^2 \dots$   
Finally, find the average of all these, and take the square root.  
For example, to find the standard deviation of the numbers 40,60,54 and 70: the average is 56. The differences are: 16, 4, 2, 14. Their squares: 256, 16, 4, 196. The average: 118 and the standard deviation is 10.86.  
Write a program that will calculate the standard deviation of a set of 20 marks.

## 2.5 Program De-bugging, Verification, and Documentation

---

These skills and habits will be practiced throughout the other sections of the course. It is good practice to:

- add simple internal documentation to identify the author, the date created, and the name of the program;
  - run the program and observe and record that there is no difference in the output between the program without documentation and the program with documentation;
  - Especially when programs get longer, discuss the use and benefits of internal documentation.
- 
- Debugging;
  - Syntax errors are listed by the compiler
  - Logical errors are more difficult to diagnose:
  - List several skills one can use, when a program does not work:
    - print out intermediate values of variables.
    - test the program in sections.
    - use simple test inputs

## 2.6 Team Programming Project (suggestions)

4 days

- 2.6.1 Write a program that lets you write the following game:  
A person starts with 50 points and must find a secret letter of the alphabet. By typing any word, the program will respond with the number of occurrences of the secret letter. If the letter does not occur in your word, you lose 10 points. If the letter occurs at least once, but less than 5 times, you lose only 5 points. If the letter occurs 5 times or more, the program halts and you win. If you get less than 0 points, you lose.  
There are many additional features that may be added.

Example: the secret letter is: F

```
Would you like to play "find the secret letter"? y
You have 50 points, what is the word? Maryhadalittlelamb
That word has 0 secret letters.
You have 40 points, what is the word? Herfleecewaswhiteassnow
That word has 1 secret letters.
You have 35 points, what is the word? Feefiefoefum
That word has 4 secret letters.
You have 30 points, what is the word? ffffffff
Congratulations, you have won.
```

- 2.6.2 Write a program where you have listed 20 words in an array of String. The program will randomly select one of the words and let you guess the word by buying a letter, similar to "wheel of fortune". If you run out of points before the word is guessed, you lose.  
There are many additional features that may be added; If you make the list longer, or use phrases instead of single words, it would make the game more challenging.

Assume that the word is: "MYSTERY"

```
You have 7 points, and the word is  *
please enter a letter:  E          *
You have 7 points, and the word is  **
please enter a letter:  N          **
You have 6 points, and the word is  ***
please enter a letter:  A          ***
You have 5 points, and the word is  ****
please enter a letter:  O          ****
You have 4 points, and the word is  *****
please enter a letter:  I          *****
You have 3 points, and the word is  *
please enter a letter:  Y          *
You have 3 points, and the word is  **
please enter a letter:  S          **
You have 3 points, and the word is  ***
please enter a letter:  T          ***
You have 3 points, and the word is  ****
please enter a letter:  R          ****
You have 3 points, and the word is  *****
please enter a letter:  M          *****
You have 3 points, and the word is  MYSTERY
```

CONGRATULATIONS, you have guessed the word.

- 2.6.3 An interesting game with dice between two players goes as follows: Each player will try to increase her total set of points, and the first one to reach 100 points wins. Players alternate turns. During a turn, a player may roll as many times as she pleases, and each sum of the pair of dice will be added to her total. However, if the player rolls doubles (i.e. the same value on each die) then that player loses all the points gained during that turn and must pass the dice to her opponent. If the player passes the dice to her opponent before she rolls doubles, her points will be permanently added to her total. Write a program to simulate this game.

# Unit 3: Problem Solving with Procedures and Functions

Time: 20 days

## Unit Description

This unit focuses on program modularity and career exploration. Students write programs using existing sub-programs and then progress to writing programs including their own sub-programs. They also explore careers in computer studies and develop skills in program modularity (e.g., a program to encrypt/decrypt a passage of text using substitution encoding).

### 3.1 The Software Development Process

1 day

#### Teaching points:

The stages of development

Emphasize the strategy of breaking a problem down into smaller manageable units.

While introducing methods, leave parameters and functions to another day

Teach the difference between local and global difference

### 3.2 Investigating Math and Text Subroutines

5 days

Teaching points: parameters, void, return

- 3.2.1 Write a method **double average(double a, double b)** that will return the average of two numbers. Then write a program to test it.
- 3.2.2 Write a method **double area (double a, double b, double c)** that will return the area of triangle ABC. Use Heron's formula. If the 3 sides cannot form a valid triangle, return -1. Then write a program to test it.
- 3.2.3 Write a method **double distance (double x1, double y1, double x2, double y2)** that will return the distance between two points in cartesian 2-space. Then write a program to test it.
- 3.2.4 Write a method **double circleArea(double r)** that will return the area of a circle of radius r. Then write a program to test it.
- 3.2.5 Write a method **double sphereVolume(double r)** that will return the volume of a sphere of radius r. Then write a program to test it. Use  $V = \frac{4}{3}\pi r^3$
- 3.2.6 Write a method **double interest(double P, double r, int n)** that will return the total amount of compound interest one will get, by lending out a principal amount, P at percentage rate r per month, compounded monthly, for n months. Write a program that will test the method. Use the formula:  $I = P \left(1 + \frac{r}{100}\right)^n - P$

- 3.2.7 Write a method **double toYearlyEquivalentRate(double r, int n)** that will convert a yearly percentage rate compounded n times to its equivalent rate compounded once per year. 12% pa compounded daily would really be equivalent to 12.747%. i.e. `toYearlyEquivalentRate(12, 256) = 12.747`. Write a program that will test the method. Use the formula:  $newRate = 100 \left( 1 + \frac{r}{100n} \right)^n - 100$
- 3.2.8 Write a method **int sumDiv(int n)** that will return the sum of the divisors of n and write a program to test it. Include in your sum the number 1, but not the number n itself. Note that `sumDiv(10) = 1+2+5 = 8`, `sumDiv(25) = 1+5 = 6`, and `sumDiv(103) = 1`. If `n < 1` return 0
- 3.2.9 Write a method **int numDiv(int n)** that will return the number of divisors of n and write a program to test it. If `n < 1` return 0.
- 3.2.10 Write a method **int gcd(int a, int b)** that will return the greatest common divisor (see program 2.4.9) If one of a,b is 0 then return a+b. Write a program to test the function
- 3.2.11 Write a method **int lcm(int a, int b)** that will return the lowest common multiple. Write a program to test the function.
- 3.2.12 Write a method **long fibonacci(long n)** that will return the nth fibonacci number. Write a program to test the function.
- 3.2.13 Write a method **boolean isEven(int n)** that will return true or false, whether n is an even number. Write a program that will test the function.
- 3.2.14 Write a method **boolean isPerfect(int n)** that will return true or false, whether n is a perfect number. A perfect number is a number whose sum of divisors is equal to the number itself. (e.g.: 6 and 8128 are a perfect numbers) Write a program that lists all the perfect numbers less than 10,000.
- 3.2.15 Write a method **boolean isPrime(int n)** that will return true or false, whether n is a prime number or not. A prime number is a number that have no divisors except 1 or the number itself. (or whose sum of divisors is 1) Write a program that will list the first 100 prime numbers.
- 3.2.16 Write a method **boolean hasFriend(int a)** that will return true or false, whether a has a friend or not. Two numbers a and b are friendly, if the sum of the divisors of a is b and the sum of divisors of b is a. Write a program that will list the friendly number pairs less than 10,000. Note that perfect numbers do not have friends. Examples of friendly numbers are: 220 and 284; 17296 and 18416

- 3.2.17 Write a method **String reverse (String word)** that reverses all the letters of word and returns it. Write a program that tests the method.
- 3.2.18 Write a method **String monthName (int m)** that returns the name of the month . monthName(5) would return "May". Any invalid input should return a string containing one blank character (" "). Write a program that tests the method.
- 3.2.19 Write a method **int dayYear (int day, int month)** that will return the day of the year. dayYear(15,5) refers to May 15, which is day 135. (Ignore leap year cases) Write a program that will test the method.
- 3.2.20 Write a method **String zodiac (int x)** that will return the name of the zodiac sign for that date. Use the method dayYear() and zodiac() to write a program that will let you test any given date.
- 3.2.21 Write a method **String primeMinister (int year)** that will return the name of the person who was prime Minister on Jan 1 of that year.

*John A. Macdonald (1867-73)*  
*Alexander Mackenzie (1873-1878)*  
*John A. Macdonald (1878-1891)*  
*Joseph Abbott (1891-1892)*  
*John Thompson (1892-1894)*  
*Mackenzie Bowell (1894-1896)*  
*Charles Tupper (1896)*  
*Wilfrid Laurier (1896-1911)*  
*Robert Borden (1911-1920)*  
*Arthur Meighen (1920-21)*  
*William L.M. King (1921-1926)*  
*Arthur Meighen (1926)*  
*William L.M. King (1926-1930)*  
*Richard Bennett (1930-1935)*  
*William L.M. King (1935-1948)*  
*Louis St.Laurent (1948-1957)*  
*John Diefenbaker (1957-1963)*  
*Lester B.Pearson (1963-1968)*  
*Pierre E.Trudeau (1968-1979)*  
*Joe Clark (1979-1980)*  
*Pierre E.Trudeau (1980-1984)*  
*John Turner (1984)*  
*Brian Mulroney (1984-1993)*  
*Kim Campbell (1993)*  
*Jean Chretien (1993-present)*

Note that Tupper, Meighen, Turner and Campbell will not appear in your subroutine.

### 3.3 Exploring Careers in Computing

3 days

(taken from "profiles")

#### Description

Students investigate career paths leading to information systems and computer science careers. They research the educational requirements of a chosen career and look into the availability of programs in colleges, universities, and private institutions that will help them meet those requirements. Students also investigate apprenticeship and co-op programs.

### 3.4.1 Programming with Subroutines (graphics)

6 days

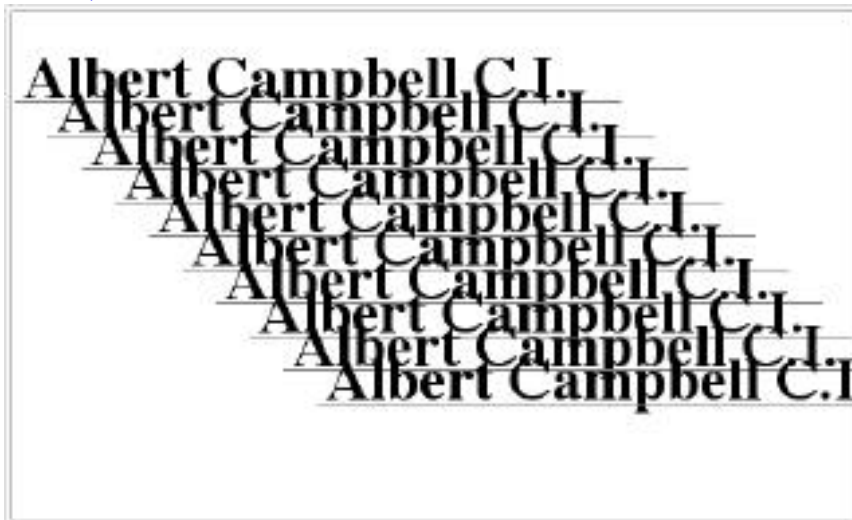
#### Teaching points:

Introduce Applets, the `init()` and `paint()` methods.

If your software has no applet viewer, use an HTML page with the `<applet>` tag introduce The Graphics class with its methods, `drawString()`, `setColor()`, `drawLine()`, `drawOval()`, `drawPolygon`, `drawRect()`, `fillOval()`, `fillPolygon`, `fillRect()`,

3.4.1.1 Write a program that will print your school's name and address in the middle of a 500x300 applet.

3.4.1.2 Write a program that prints your school name as follows, using a loop, the name in red, the underline in blue.

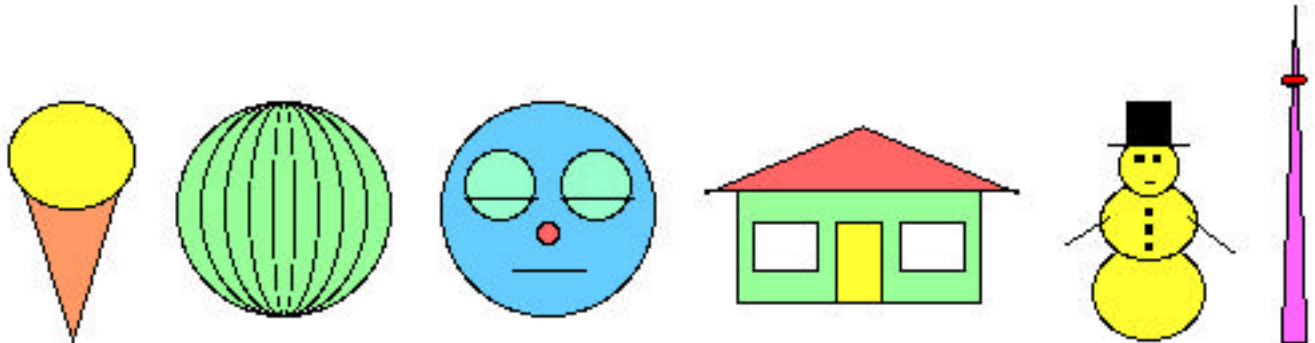


3.4.1.3 Write a program where your name is repeated in successively larger letters:

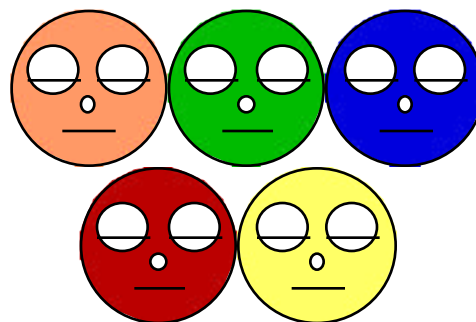
```
Albert Campbell C.I.  
Albert Campbell C.I.  
Albert Campbell C.I.  
Albert Campbell C.I.  
Albert Campbell C.I.  
Albert Campbell C.I.  
Albert Campbell C.I.
```

3.4.4 Create a method `Color randColor()` that will generate a random colour, and use it to modify the previous applet

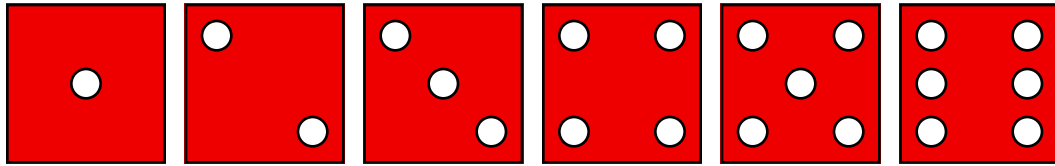
3.4.1.5-10 Write a program that lets you draw the following or similar shapes:



3.4.1.11 Write a method that will draw a face, leaving with the face colour as an input parameter. Then write an applet that will draw several faces on the screen:



3.4.1.12 Write a method **void oneDie(int x, int y, int value)** that will draw a die with the upper left corner at position (x,y) and showing the value "value"



then test the method by randomly generating 2 dice in the middle of the field, together with their sum, using one red and one blue die

### 3.4.2 Programming with Subroutines (GUI)

5 days

#### Teaching points:

Introduce the action method,  
Introduce textfields, labels, buttons

- 3.4.2.1 Write a program that lets you enter a celsius degree measure and that will convert it to fahrenheit degrees.
- 3.4.2.2 Write a program that lets you enter an integer and that finds the nearest prime number larger and smaller than itself. If the number is a prime number, only print the number itself.
- 3.4.2.3 Use a label and three textboxes in a program that lets you enter the three sides of a triangle and that prints out (if the three sides can form a triangle) the perimeter, the area.

enter the 3 sides of a triangle

- 3.4.2.4 Use a label and three textboxes in a program that lets you enter the three sides of a triangle and that prints out (if the three sides can form a triangle) the values (in degree measure) of the three corresponding angles, using the cosine law  
The angle A, (contained by sides b,c and opposite side a) can be found using the formula:  $A = \cos^{-1} \frac{b^2 + c^2 - a^2}{2bc}$ , where C must next be converted to degree measure:  $A *= 57.2958$ . the java function `Math.acos()` corresponds to  $\cos^{-1}()$   
Create the method **double cosLawSSS(double b, double a, double c)** that will return the angle opposite side a (the middle parameter)

- 3.4.2.5 Use a label and three textboxes in a program that lets you enter two sides and a contained angle, and that will print side a, opposite angle A and the angles B and C, opposite sides b and c respectively.  
Create the method **double cosLawSAS(double b, double A, double c)** that will return the side opposite angle A (the middle parameter) Use the cosinelaw formula:  $a = \sqrt{b^2 + c^2 - 2bc \cos(A)}$ .  
Don't forget to convert angle A to radian measure first:  $A /= 57.2958$ .

angle A      side b      side c

- 3.4.2.6 Write a program that lets you guess a number between 1 and 100 (see 2.4.4)
- 3.4.2.7 Write a program that uses one button; "roll"  
When you press the button, two random dice will show (use the method of 3.4.1.12)

- 3.4.2.8 Write a program that uses 3 radio buttons called "rock", "scissors", "paper" and a button called "GO". Write a program where you play the computer. When you make a choice, the computer too makes a random choice. When you press "GO" the computer will display the result. Note that : Rock wins over Scissors; Scissors wins over Paper and Paper wins over Rock. Choosing the same implement results in a draw.
- 3.4.2.8 Write a program that uses 6 buttons: 1,2,3,4,5,6. When you press an button, your (red) die will show with the appropriate number. the program (the computer) must roll one (blue) die. If your die is the same or differs only by 1 from the computer's die, then you win, otherwise you lose. Note, that 6 and 1 are considered adjacent and "differ by 1" (given any die that is cast, there are three opportunities to win, and three opportunities to lose)

# Unit 4: Information Storage and Related Issues

Time: 8 days

## Unit Description

This unit focuses on data storage and manipulation. Students examine issues surrounding privacy, security, and ethical use of information. They also write programs that input data from existing files, process the data, and create files for external data storage, following an appropriate problem-solving model (e.g., Create a data file containing employee information including hours worked and rate of pay. Read from the file, compute, display, and write to a new file the gross pay for each employee.).

## 4.2 Reading Data Files

4 days

### Teaching points

Input data from an existing file (see zodiac.txt and premiers.txt found elsewhere in this paper)

introduce `BufferedReader` and `FileReader` and `StringTokenizer`

- 4.2.1 Write a program that reads the file `zodiac.txt` located on the a-drive and prints the first 9 lines on the console screen.  
(you will find the `zodiac` file in the appendix)
- 4.2.2 **zodiac.txt** contains 12x9 lines. Each group of 9 lines contain in order: Name of the house, dates, personality, colours, stones, metal, trees, flowers and the ruler of the house.  
Write a program that will list the name of the house and its corresponding dates
- StringTokenizer()**
- 4.2.3 Write a method **String chopWords(String line, int num)** which will insert the special character '\n' in the space every "num" words. The effect is, that when the line is printed, it will only print "num" words per line.  
`System.out.println(chopWords("Mary had a little lamb its fleece was white as snow",5))` would print out: 

```
Mary had a little lamb
its fleece was white as
snow
```

  
Write a program that lets you input a line of words followed by a number, and that will format the line using `chopWords()`.
- 4.2.4 Write a program that will let you enter a number, 1 -12 and that will print out the Zodiac name, its dates and the personality description, 8 words per line. Use `chopWords()`.
- 4.2.5 Write a method `String lister(String line, int columns)` that will insert the special character "\n" into the space between some words, in such a way, that when printed, the entire text of the string "line" will fit in "columns" spaces. Use this method to write a program that will print out a zodiac name with the personality, given a number input (1-12)

### 4.3 Creating Data Files

3 days

**Teaching points:** FileWriter, PrintWriter

Applets have no writing privileges to files... why not?

- 4.3.1 Write a program that lets your enter 10 of your friends' names into the text file named a:friends.
- 4.3.2 Create a file containing all the Prime Ministers of Canada, together with the year when they were first elected. Each Prime Minister should take up 3 lines: (1) last name; (2) first name(s); (3) election year. Let the last line contain the special word: "EOF"
- 4.3.3 Write a program that will read all the data from the PrimeMinister file and create two new files: The first file will contain all the prime ministers who were elected in the 19th century and the second file will contain those who were elected in the 20th century. Note that you must read the file until you reach the line containing "EOF"
- 4.3.4 Write a program that will read all the data from the zodiac file and creates a new file, that will only contain the zodiac name, its dates and the ruler of the house.
- 4.3.5 Write a program that will let you enter a number (1 = personality, 2 = colours, 3 = stones, 4 = metal, 5 = trees, 6 = flowers and 7 = the ruler of the house) and that will create a new file that contains the zodiac name, its dates and the requested quality of the house for all twelve houses.

# Unit 5: Using Data Structures

Time: 18 days

## Unit Description

This unit focuses on the programming techniques required to store and manipulate data and to solve problems through the development of a database. Each activity develops knowledge and skills that students apply in the culminating challenge of this unit: to develop a database for a school team (e.g., the hockey team or similar organization, consisting of personal data such as player name, position played, jersey number, phone number, goals, and assists). Students examine the structuring of one- and twodimensional arrays and how data is represented and stored in these structures. They write programs that create lists and tables of data, manipulate the data, and output the result. Sorting and searching techniques are also applied.

## 5.2 Data in Lists

3 days

### Teaching points:

manipulating arrays by means of for loops

It is a good practice in all data files, to have the first line contain a number that represents the total number of lines in the text file, or to have the last line contain a special word such as "<EOF>" to signal the end of the file, especially when the length is likely to change.

Note, this is a natural place to teach students about creating Objects and records, but this is to be left until grade 12.

- 5.2.1 Write a program that will generate a list of n random numbers and saves it in a text file: The first line of the file will contain the number n. Let n be the number 1000, let the random numbers be integers between 1 and 100 (inclusive)
- 5.2.2 Write a program that will load the number file from 5.2.1 into an array of integers and that will let you enter a number between 1 and 100. It will then print out how often that number occurs in the file.
- 5.2.3 Modify 5.2.2 to let you print out how many numbers are in a given range of values.
- 5.2.4 Write methods **void loadSet (); void saveSet (); void replaceOne (int newNum, int where)** that will load the file from 5.2.1 into a global array of numbers; saves it again to the file; and replace the number in position "where" by the value of "newNum". Use these methods in a program that will let you change the value of the number in a specific location within the file.
- 5.2.5 Prepare the random number file of 5.2.1 by replacing one number by a value larger than 100. Then write a program that will locate the position of that number in the file. (Provide for the possibility that the number cannot be found)

5.2.6 Prepare a data file that contains only the last names of the Prime ministers. Then write a program that lets you load all the prime ministers (last names only) from the textfile into an array and that lets you add a new name to the end of the array. Your program must print the list before and after adding the name (to the end of the list.

Therefore, write 4 methods: **void loadPM(); void savePM(); void printPM()** and **void addPM(String name)**.

5.2.7 Modify the 5.2.6 so that you can delete a name as well as adding one. Note that in your list of names all prime ministers who come after the deleted one must move down (or the prime minister who is in the last spot must move into the vacated spot. )

### 5.3 Relating Lists

2 days

5.3.1 Write a method **void houseList( int listItem)** that loads the titles of all 12 houses into the global String array **house[12]** and the appropriate property into the global String array **property[12]**. For example **houseList(2)** would load the house names into **house[12]** and the personality descriptions into **property[12]**. Note that 1 stands for dates, 2 for personalities, 3 for colours, etc. Then write a program that will let you ask for an appropriate property, and list all 12 signs with that property. (Compare with problem 4.3.5)

5.3.2 Write a program that will modify the program in 5.3.1 to print out the complete data set of a requested house.

5.3.3 Modify 5.3.2 to enter the name of the house (virgo, scorpio,...) and that will print out the appropriate data

5.3.4 Modify 5.3.2 to enter the date and finds the appropriate data. You may wish to convert dates to "day of year" (see 3.2.19)

### 5.4 Data in Tables (two dimensional arrays)

2 days

5.4.1 The Pascal triangle turned on its side looks as follows:

1	1	1	1	1	1	1	1	1	1
1	2	3	4	5	6	7	8	9	10
1	3	6	10	15	21	28	36	45	55
1	4	10	20	35	56	84	120	165	220
1	5	15	35	70	126	210	330	495	715
1	6	21	56	126	252	462	792	1287	2002
1	7	28	84	210	462	924	1716	3003	5005
1	8	36	120	330	792	1716	3432	6435	11440
1	9	45	165	495	1287	3003	6435	12870	24310
1	10	55	220	715	2002	5005	11440	24310	48620

The first row are the numbers in  $p[0][0]$ ,  $p[0][1]$ ,  $p[0][2]$ , ...,  $p[0][9]$   
 The familiar Pascal triangle is the triangle of numbers with  $p[0][0]$  as vertex and the set of numbers  $p[0][9]$ ,  $p[1][8]$ ,  $p[2][7]$ ,  $p[3][6]$ , ...,  $p[9][0]$  as base.  
 Note that  $p[0][y]$  is always 1, and that any other number is obtained by adding two of its neighbours: the one above and the one to its left.  
 The number 15 for example is  $5+10$  (or  $10+5$ , depending on which 15 you found)  
 Write a program that will prints this array of numbers

5.4.2 Modify the program above by printing only the triangle, that is, the first 10 numbers, followed by the next 9, then the next 8 and so on, until it only prints one number from the last row of the array.

5.4.3 The pascal triangle can also be written as follows:

1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0
1	3	3	1	0	0	0	0	0	0
1	4	6	4	1	0	0	0	0	0
1	5	10	10	5	1	0	0	0	0
1	6	15	20	15	6	1	0	0	0
1	7	21	35	35	21	7	1	0	0
1	8	28	56	70	56	28	8	1	0
1	9	36	84	126	126	84	36	9	1

Here the entire first row is 0 and the first column is 1 initially.  
 All the other numbers are obtained by adding the number directly above it and the number diagonally to the left and above  
 Write a program that will print out this pascal triangle, but do not print out the cells that contain 0.

5.4.4 Modify program 5.3.3 to read all the data from the zodiac.txt file into the array `zodiac[12][9]`

## 5.5 **Sorting Data** **6 days** **Teaching points**

Only teach the simple sorts: bubble, insertion and selection sort

5.5.1 Write a program that fills an array of 20 with a set of random integers between 1 and 1000, prints out the list on one line, and prints the largest of the set. on the next line.

5.5.2 Write a program that fills an array of 20 with a set of random integers between 1 and 1000, prints out the list on one line, and prints the smallest of the set. on the next line.

5.5.3 Write a program that fills an array of 20 with a set of random integers between 1 and 1000, prints out the list on one line, and prints the largest of the set. on the next line, together with its position (a number between 1 and 20 inclusive)

- 5.5.4 Write a program that fills an array of 20 with a set of random integers between 10 and 99, prints out the list on one line, and underlines the two-digit largest number with the symbol "==".
- 5.5.5 Write a method **void swap (int a, int b)** that will switch the values of the a<sup>th</sup> and b<sup>th</sup> array elements. Then write a program that will find the largest of a set of 20 numbers and place it in position 0.
- 5.5.6 Selection Sort: Imagine a random set of numbers in a global array, say list[0],..., list[n-1]. Write a method **void smallest(int a, int b)** that uses the swap routine to move the smallest number of the subset list[a], ..., list[b] into list[b]. Then write a program that sorts the entire array by successively using this method on the subsets list[i], ..., list[n-1], where i runs from 0 to n-2.
- 5.5.7 In groups of two or three, simulate the selection sort with a set of 10 sheets each containing a random number.
- 5.5.8 In groups of two or three, simulate the bubble sort.
- 5.5.9 Write a program that uses the bubble sort to sort a set of random numbers.
- 5.5.10 Write a method void shiftRight (int a, int b), where a<b that shifts all the values in the global array list[ ] one position to the right. The contents on list[b] will move to list[a]. Then modify the program in 5.5.5 for insertion sort:
- 5.5.11 In groups of two or three, simulate insertion sort. First modify the program to print out each pass. Then identify in each pass, the values of a and b, by underlining the subset from a to b:

<u>69</u>	92	25	24	55	97	84	5	32	95	48	95	27	48	12	70	95	86	29	92
69	<u>92</u>	25	24	55	97	84	5	32	95	48	95	27	48	12	70	95	86	29	92
<u>25</u>	<u>69</u>	<u>92</u>	24	55	97	84	5	32	95	48	95	27	48	12	70	95	86	29	92
<u>24</u>	<u>25</u>	<u>69</u>	<u>92</u>	55	97	84	5	32	95	48	95	27	48	12	70	95	86	29	92
24	25	<u>55</u>	<u>69</u>	<u>92</u>	97	84	5	32	95	48	95	27	48	12	70	95	86	29	92
24	25	55	69	92	<u>97</u>	84	5	32	95	48	95	27	48	12	70	95	86	29	92
24	25	55	69	<u>84</u>	<u>92</u>	<u>97</u>	5	32	95	48	95	27	48	12	70	95	86	29	92
<u>5</u>	<u>24</u>	<u>25</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>97</u>	32	95	48	95	27	48	12	70	95	86	29	92
5	24	25	<u>32</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>97</u>	95	48	95	27	48	12	70	95	86	29	92
5	24	25	32	55	69	84	92	<u>95</u>	<u>97</u>	48	95	27	48	12	70	95	86	29	92
5	24	25	32	<u>48</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>95</u>	<u>97</u>	95	27	48	12	70	95	86	29	92
5	24	25	32	48	55	69	84	92	95	<u>95</u>	<u>97</u>	27	48	12	70	95	86	29	92
5	24	25	<u>27</u>	<u>32</u>	<u>48</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>97</u>	48	12	70	95	86	29	92
5	24	25	27	32	48	<u>48</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>97</u>	12	70	95	86	29	92
5	<u>12</u>	<u>24</u>	<u>25</u>	<u>27</u>	<u>32</u>	<u>48</u>	<u>48</u>	<u>55</u>	<u>69</u>	<u>84</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>97</u>	70	95	86	29	92
5	12	24	25	27	32	48	48	55	69	<u>70</u>	<u>84</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>95</u>	<u>97</u>	86	29	92
5	12	24	25	27	32	48	48	55	69	70	84	<u>86</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>95</u>	<u>97</u>	29	92
5	12	24	25	27	<u>29</u>	<u>32</u>	<u>48</u>	<u>48</u>	<u>55</u>	<u>69</u>	<u>70</u>	<u>84</u>	<u>86</u>	<u>92</u>	<u>95</u>	<u>95</u>	<u>95</u>	<u>97</u>	92
5	12	24	25	27	29	32	48	48	55	69	70	84	86	92	<u>92</u>	<u>95</u>	<u>95</u>	<u>95</u>	<u>97</u>

- 5.5.12 Modify selection sort to sort Strings. Then load the prime minister's data from your text file into an array and sort them.
- 5.5.13 Load the zodiac text file into a 12x9 array and modify the selection sort to sort the file.

## 5.6 Searching Lists and Tables

3 days

**Teaching points:** Linear and binary searches

Discuss the advantages of both.

Using the best strategy of the Hilo game, #2.4.4, demonstrate that the maximum of number of tests that needs to be performed is equivalent to repeatedly dividing the total number of the search set in half. (If the number of elements to be searched is  $n$ , the number of tests is never more than  $\log_2 n$ )

- 5.6.1 Generate a set of 1000 random numbers between 1 and 1000 and save the set in the text file: **random.txt**.  
Then write a program that will first let you load the set of numbers into an array and secondly lets you enter a number for testing to see if that number is in the file. Use a method **int positionOf(int number)** that will return the position of the first occurrence of "**number**" in the list of 1000. If the number is not there, then return the number **-1**.
- 5.6.2 Write a program that will sort the set of numbers in **random.txt** from smallest to largest, and stores them in the file **sorted.txt**. Write a program that modifies 5.6.1: Change the linear search so that the program stops as soon as the data in the file is larger than the number you are looking for.
- 5.6.3 Use a set of 10 random numbers on paper, placed in numeric order, to simulate a binary search. Write down each number that has been sampled, in the process of finding the position of a number.
- 5.6.4 Write a program that will search the file **sorted.txt** using a binary search.
- 5.6.5 Modify program 5.6.4 to read the prime minister's file (see 3.2.21) and search it by last name, using a linear search.

# Unit 6: Putting It All Together

Time: 16 days

## Unit Description

This unit is the culminating challenge for applying knowledge and skills in an integrated and meaningful task. Students follow the software design life cycle to find the best solution to a challenge (e.g., a movie reservation system), demonstrating the mastery of course expectations. The teacher should choose the challenge with students to allow students to express their creativity while at the same time demonstrating knowledge and skills. Students examine the effect and influence on society of emergent technologies.

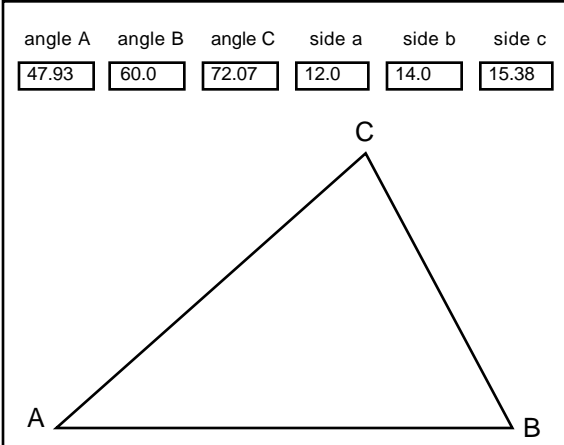
**Each of the above activities will be stages of the same problem. The following are sample projects students may be working on in groups.**

- 6.1 You are to create an applet, that lets you enter 6 numbers representing the sides and angles of a triangle: Three of these numbers are positive, and three are 0. Your program should (using trigonometry) draw a properly scaled diagram of the triangle, and replace the zeroes with the calculated values.

angle A	angle B	angle C	side a	side b	side c
0	60	0	12	14	0

angle A	angle B	angle C	side a	side b	side c
47.93	60.0	72.07	12.0	14.0	15.38



- 6.2 You are to create an applet that makes use of a database (or table) you have prepared of a list of your favourite hockey, baseball, or basketball team, which contains all the players' statistics. Your program should let you enter a name (or part name) and print that player's information.
- 6.3 You are to write an applet that presents a multiple choice test on a topic of your choice. There should be at least 20 questions, with 5 choices, and only one choice being correct. The questions and choices should be presented in random order.

- 6.4 Create an program that lets you maintain a database such as in 6.2. The user should be able to:
- list all the players' names
  - list all the information on one player, given the last name
  - list all the information on one player, given a player's unique code
  - add the data of a new player
  - delete the data of a retired player
  - edit the information of a player
- 6.5 Create an applet that lets you simulate a "wheel of fortune" game, based on the names of popular movies (for example). Modify the game so that only one person can play.

## Appendix: Zodiac.txt

### AQUARIUS

21 January -- 19 February

The Scientist of the Zodiac, Aquarius is ruled by URANUS, and its symbol is the WATER-BEARER. Its natural place is the ELEVENTH HOUSE, the house of Friendships and Experimentation. Revolutionary Aquarius is an AIR sign, idealistic and inventive. Aquarius often mood swings, optimistically reforming the social structure one minute, then despondent at the impossibility of perfect happiness the next.

Electric blue, Shocking pink, Fluorescent hues. All the bright colours of the rainbow

Sapphire, Opal Chalcedony

Uranium

Cornus Eucalyptus

Orchids, Pelargonium Hibiscus

Uranus

### ARIES

21 March -- 20 April

The Pioneer of the Zodiac, Aries is ruled by MARS, and its symbol is the RAM. Its natural place is the FIRST HOUSE, the house of Personal Identity. Independent Aries is a FIRE sign, impulsive and energetic. Rarely looking before it leaps, Aries sometimes has a problem completing tasks, too caught up in the allure of the Next Big Thing.

Bright primary colours, particularly: Scarlet Yellow Red

Diamonds, garnets bloodstones

Iron Steel

All thorn bearing trees and shrubs

Daffodils Tulips Honeysuckle

Mars

### CANCER

22 June -- 22 July

The Home-Maker of the Zodiac, Cancer is ruled by THE MOON, and its symbol is the CRAB. Its natural place is the FOURTH HOUSE, which is concerned with Matters of Home Life. Nurturing Cancer is a WATER sign, supportive and emotional. Prone to emotional tides, a Cancer believes "a man's home is his castle" -- and cocooning a legitimate lifestyle choice.

Silver, Cerulean blue Sea green

Pearl Moonstone

Silver

Willow, Spruce Pear

Daisy Acanthus Iris

Moon

## CAPRICORN

22 December -- 20 January

The Organizer of the Zodiac, Capricorn is ruled by SATURN, and its symbol is the SHE-GOAT. Its natural place is the TENTH HOUSE, the house of Status and Career.

Ambitious Capricorn is an EARTH sign, worry-prone and steadfast. Come to Capricorn with your troubles, and be prepared for worried -- even meddling -- advice, but also a strong shoulder to lean on.

Deep green, Brown Dark grey

White onyx Jet Sapphire

Lead

Pine Elm Popular

Pansies Ivy Sedum

Saturn

## GEMINI

22 May -- 21 June

The Communicator of the Zodiac, Gemini is ruled by MERCURY, and its symbol is the TWINS. Its natural place is the THIRD HOUSE, the house of Communication. Curious Gemini is an AIR sign, analytical and ever changing. Often times overly intellectual and unstable, Gemini reserves the right to change its mind, often "forgetting" to communicate the unpleasant.

White Saffron Yellow Black and white contrasts

Crystal Topaz Beryl.

Mercury

Nut-trees Rowans Beech

Lilly-of-the-valley, Lavender Rock-roses

Mercury

## LEO

23 July -- 23 August

The Entertainer of the Zodiac, Leo is ruled by THE SUN, and its symbol is the LION. Its natural place is the FIFTH HOUSE, the house of Children and Pleasure. Expressive Leo is a FIRE sign, creative and grandiose. Keep Leo at the center of attention, and you'll both be happy.

Gold

Ruby Diamond Hyacinth

Gold

Palm Citrus fruit trees and rhododendron

Sunflower Potentilla

Sun

## LIBRA

24 September -- 23 October

The Diplomat of the Zodiac, Libra is ruled by VENUS, and its symbol is the SCALES. Its natural place is the SEVENTH HOUSE, the house of Partnerships. Peace loving Libra is an AIR sign, tactful and decisive. Depending on how the scales tip, Libra can be a brilliant creator -- or a babbling madman.

Pale greens Pretty pinks Pastel shades Clear blue

Diamond, Cornelian Lapis lazuli.

Copper Aluminum

Acer Laburnum Plum trees

Cyclamen Clematis

Venus

## PISCES

20 February -- 20 March

The Mystic of the Zodiac, Pisces is ruled by NEPTUNE, and its symbol is the FISH. Its natural place is the TWELFTH HOUSE, the house of Sorrow and Karma. Intuitive Pisces is a WATER sign, sensitive and adaptable. A chameleon, insecure Pisces can powerfully attain the attributes of any other sign -- all to help it fit in

Violet White Blue grey

Amethyst, Chrysolite and Topaz

Tin

Fig Peaches

Water lilies, Delphinium Dahlias

Neptune

## SAGITTARIUS

23 November -- 21 December

The Gambler of the Zodiac, Sagittarius is ruled by JUPITER, and its symbol is the ARCHER. Its natural place is the NINTH HOUSE, the house of Travel and Philosophy. Expansive Sagittarius is a FIRE sign, goal oriented and optimistic. Blunt Sag is always willing to help, particularly if also promoting its own advancement.

Warm brown Autumnal tints Brilliant blue Dark blue

Carbuncle Topaz Hyacinth

Tin

Lime Mulberry Oak

Carnations

Jupiter

## SCORPIO

24 October -- 22 November

The Seducer of the Zodiac, Scorpio is ruled by PLUTO, and its symbol is the SCORPION. Its natural place is the EIGHTH HOUSE, the house of Passions -- Sex, Birth, and Death.

Intense Scorpio is a WATER sign, secretive and magnetic. Strong willed Scorpio's heart is closed to all but an intimate few -- but its still waters run deep.

Dark red Brown Black

Topaz, Malachite Jasper

Sodium

Yew Cedar Acacia

Azalea Gladiolus Hellebore

Pluto

## TAURUS

21 April -- 21 May

The Builder of the Zodiac, Taurus is ruled by VENUS, and its symbol is the BULL. Its natural place is the SECOND HOUSE, the house of Possessions. Productive Taurus is an EARTH sign, reliable and materialistic. With a fine aesthetic sense and stubborn will to have things its own way, Taurus invariably surrounds itself with material beauty -- be it artistic or natural.

Cream, Madonna blue Spring green

Emeralds, Coral Lapis lazuli

Copper

Ash Cypress Apple

Rose Poppy Violets Foxgloves

Venus

## VIRGO

24 August -- 23 September

The Perfectionist of the Zodiac, Virgo is ruled by MERCURY, and its symbol is the VIRGIN. Its natural place is the SIXTH HOUSE, the house of Work and Duty. Practical Virgo is an EARTH sign, detail oriented and obsessive. Clean-freak Virgo wants a tidy desktop -- and will do anything to get it.

Light blue Pure white Pastel colours in neat patterns

Pink jasper, Hyacinth Topaz

Platinum

Cherry Cotton tree Birch

Alpine Rock plants

Mercury