

A blue parallelogram and a light green parallelogram are positioned in the top-left corner of the slide. The blue shape is partially behind the green one. Both shapes are oriented diagonally, with their longer sides running from the top-left towards the bottom-right.

Programming Techniques COSC1284/2010

Tutorial 5

Exercise 5.1 Solution

```
/*  
  
    Expression                                Result  
    yes == no || grade > amount              true  
    amount == 40.0 || 50.0                   Error  
    hiVal != loVal || loVal < 0               true  
    True || hello.length() > 0                Error  
    hello.isEmpty() && yes                    false  
    grade <= 100 && !false                     true  
    !yes || no                               false  
    grade > 75 > amount                       Error  
    amount <= hiVal && amount >= loVal        true  
    no && !no || yes && !yes                  false  
  
*/
```



Agenda

- Tutorial/Lab
 - Read chapter 6 from the textbook
 - Discuss the concepts with your tutor and fellow classmates
 - Complete chapter 6 - Exercises 1 - 3
 - Attempt on your own
 - Complete chapter 6 - Exercises 4 - 6
- Note: Please refer to tutorial 4 for online instructions.



Loops and Strings

- We come to the point where need to repeat our code, such as sorting, search, displaying a picture (number of pixels), showing a movie (number of frames) and many more examples.
- Essentially, there are three basic and common ways to loop our code, the for loop, the while loop and the do while loop.
- The main difference between the loops; the for loop generally loops with a known end point, where a while loop has an unknown endpoint.
- The majority of the programs that we have looked at involves numbers, such as conversions of temperatures or dates, it would be advantageous if we could representative text too.
- We accomplish this by using a data type called String, which is basically made up of a series of characters as well as a number of methods that we can use with Strings.



Exercise 6.1

```
public static void main(String[] args) {  
    loop(10);  
}  
  
public static void loop(int n) {  
    int i = n;  
    while (i > 1) {  
        System.out.println(i);  
        if (i % 2 == 0) {  
            i = i / 2;  
        } else {  
            i = i + 1;  
        }  
    }  
}
```

1. Draw a table that shows the value of the variables i and n during the execution of loop. The table should contain one column for each variable and one line for each iteration.
2. What is the output of this program?
3. Can you prove that this loop terminates for any positive value of n ?



Exercise 6.2

- Let's say you are given a number, a , and you want to find its square root. One way to do that is to start with a rough guess about the answer, x_0 , and then improve the guess using this formula:

$$x_1 = (x_0 + a/x_0) / 2$$

- For example, if we want to find the square root of 9, and we start with $x_0 = 6$, then $x_1 = (6 + 9/6) / 2 = 3.75$, which is closer. We can repeat the procedure, using x_1 to calculate x_2 , and so on. In this case, $x_2 = 3.075$ and $x_3 = 3.00091$. So it converges quickly on the correct answer.



Exercise 6.2 (cont)

- Write a method called `squareRoot` that takes a `double` and returns an approximation of the square root of the parameter, using this technique. You should not use `Math.sqrt`.
- As your initial guess, you should use $a/2$. Your method should iterate until it gets two consecutive estimates that differ by less than 0.0001. You can use `Math.abs` to calculate the absolute value of the difference.



Exercise 6.3

- One way to evaluate $\exp(-x^2)$ is to use the infinite series expansion:

$$\exp(-x^2) = 1 - x^2 + x^4/2 - x^6/6 + \dots$$

- The i th term in this series is $(-1)^i x^{2i} / i!$. Write a method named `gauss` that takes x and n as arguments and returns the sum of the first n terms of the series. You should not use `factorial` or `pow`.