COMP 1409 – Summary of Session 4 Concepts

- Boolean operators. These allow us to combine expressions to form more complex logic operations. The boolean operators are "and" `( && )`, "or" `( || )` and "not" `(!)` These operators also result in `true` or `false`.
- Truth table. `P` and `Q` are expressions that evaluate to `true` or `false`.

P	Q	P && Q	P || Q	!Q
true	true	true	true	false
true	false	false	true	true
false	true	false	true	false
false	false	false	false	true

- Precedence of operations. See Appendix C in your text. In general, remember that expressions inside parentheses receive high precedence, the same as they do in math. `!` (not) is evaluated before `&&` (and) and `||` (or).
- Making choices. The `if` statement is used to make a decision and cause the execution of the program to branch.

```
if (boolean condition) {
   // statements to execute if the condition is true
}
else { // else is optional
   // statements to execute if the condition is false
}

/** to demonstrate */
public void demonstrateIf(int number)
{
  if(number <= 0) {
      System.out.println("number is negative or zero");
  }
  else if((number > 0) && (number < 10)){
      System.out.println("between 0 and 10");
  }
  else {
      System.out.println("number is 10 or higher");
  }
}
```

- Local variables. These are variables declared inside methods. They do not have a visibility modifier (public or private) in front.

- Scope. The scope of a variable refers to where that variable is visible within the class. Instance variables (fields) have scope throughout the entire class. Parameters have scope limited to the method or constructor. Local variables have scope limited to the block of code where they are declared.

```
public class Student
{
   private String name; // scope is the whole class

   public Student(String sName) //scope is the constructor
   {
      name = sName;
   }

   public String getName()
   {
        return name; // name visible here
        // sName not visible
   }

   public void setName(String newName) // scope is method
   {
      name = newName; // name visible here
      // sName not visible
   }

   public void showInfo()
   {
       String label = "Name is "; // scope is just here
       System.out.println(label + name); // name visible
   }
}
```

- Lifetime. The lifetime of a variable refers to when that variable holds a value. Instance variables (fields) exist while the object exists. Parameters exist while the method or constructor is executing. Local variables exist while the block of code they are declared in is executing.
- Static variables. A static variable is shared by all objects of a class.