COMP 1409 – Summary of Session 2 Concepts

- Parts of a class. An outer wrapping names the class, and a larger inner part contains the fields, constructors and methods of the class. Fields store data for the class. Constructors set up the object created from the class by initializing the data members (fields). Methods implement the behaviours of the object that is created.
- Access modifiers, private and public. Fields are defined as private to ensure that only the methods belonging to the class will have access to the data. The class itself and its methods are defined as public so that they may be referred to from outside. (A method could be defined as private and referred to only from another method of that class.)
- Constructors look like methods except that they have the same name as the class and they do not specify a return type, not even void. A constructor may or may not have one or more parameters passed to it. Java supplies a default constructor that initializes all the fields to null values but it is better to explicitly create your own to set up the object in exactly the way you want. It is the constructor that is called when you click on **`new`** in BlueJ to create an object.
- Comments are descriptions in English, meant to make code easier to understand. The class itself should be commented, as well as each field, each constructor and each method. Comments are also used within the methods to explain the logic. Three types of comments in Java:

  ```
  /*
     this is a comment spanning several lines
  */

  // this is a comment to the end of the line

  /**
  *     this is a Javadoc comment
  */
  ```

- Parameter passing, sometimes called argument passing. The signature of a method or a constructor specifies the data type to be passed to the method. This is known as the formal parameter. The actual value that is passed when the method or constructor is called is known as the actual parameter, or the argument. The formal parameter is a variable (a named place in memory) that exists only while the method is being executed. The actual parameter might be a literal value, or a variable, or the return value from a method call.
- Variable declaration tells the compiler to reserve a chunk of memory of a specific size. The variable declaration must include the data type and the name of the variable, e.g. `private int age;`
- Assignment statement is the statement that changes the value in a variable. The assignment operator (=) takes the value in the expression on the right and puts that value into the variable on the left, e.g. `counter = 0;`

- Scope refers to the degree of visibility of variables. The scope of an instance variable (field, data member of the class) is the entire class. The scope of a parameter is inside its method. The scope of a local variable is inside the braces that enclose it, e.g.

  ```
  {
       int number = 0;
       // do something with number
  }
  ```

- Accessor methods. These are methods that provide information about an object, typically by returning the contents of one of the instance variables.
- Mutator methods. These are methods that change the state of an object by changing the value in one or more of the instance variables. They typically receive parameters and contain assignment statements.