

## COMP 1409 – Summary of Session 1 Concepts

- **Class:** an abstraction that defines a type of entity, e.g. Car. A class defines fields (attributes) and methods (behaviors).
- **Object:** an instantiation of a particular class, e.g. myCar, yourCar, thisCar, thatCar. Each object holds its own data in its fields.
- **Field:** also known as instance variable. These are variables declared for the class, not inside a method. The methods defined for the class have access to the fields and can change their contents. Each object created from a class will have its own copy of each of these fields. This is where the data for the object is stored.
- **Variable:** a named memory location, used to hold a value of a particular data type.
- **Data type:** kind of data used, a way to specify how much memory to allocate. Some common data types are int (whole number, e.g. 5, 999, -34), char (single character, e.g. 'h', '9', '&'), String (group of characters, e.g. "abc", "1234", "hello there"), double (number with a decimal component, e.g. 8.98, -7654.9099), boolean (true, false). Different data types use different sized memory locations.
- **State:** the set of values of all attributes that define an object. The data values in the fields of a particular object.
- **Method:** a group of programming statements that is given a name. Methods are defined for a class. The objects of that class use the methods to access the data in their fields. (In non-OO languages methods are called subroutines, procedures, or functions.)
- **Parameter:** a value that is passed to a method inside parentheses when the method is invoked, e.g. myCar.drive(30). The value 30 is the parameter passed to the drive method belonging to the myCar object.
- **Source code:** the instructions that define the attributes and behaviors of a class. These instructions are written in a programming language, e.g. Java.
- **Compile:** an operation to translate Java instructions into machine code the computer can understand.
- **Compiler:** a program that compiles source code into machine code.
- **Return value:** a value returned by a method. This might be the contents of one of the fields, or it might be the result of some calculation or other operation. The type of the return value is part of the method signature.
- **Method signature:** also known as the method header. Information about the method, e.g. public void changeSize(int newHeight, int newWidth). This specifies a method that (1) is accessible outside the class, (2) does not return a value, (3) is named "changeSize", (4) requires two parameters of type int to be passed to it when it is invoked.
- **Void:** part of a method header to specify that the method does not return a value. If the method does return a value, the data type of the return occupies the same position in the method signature, e.g. public boolean isVisible(). This method returns either true or false when it is invoked.