# JAB Revision

The Java test will be open book – you can use your LEARN resources.

The Test classes will be provided for you.

You will have 1 ½ hours to complete the test.

**Sessions 1-9 revision:**

* **Classes and Objects**
* Create a new class (Class1)
* Create private instance variables (determine the datatypes)
* Define public constants to initialise the private instance variables
* Create a no-arg constructor – use chaining to the all-arg using this(..)
* Create an all-arg constructor
* Create any other partial-arg constructors – use chaining to the all-arg using this(..)
* Create the “setter” and “getter” methods for all the private instance variables
* Create a “toString()” method
* Create an “equals()” method
* Create an additional method for the class. A method can accept none or more parameters and can return a value
* A Test class will be provided to test all methods
* **If, if-else, nested-if, switch**
* **for, while and do while loops**
* **Aggregation**
* Create a class that uses the class created (eg Class1 above) as an instance variable
* Create one or more other instance variables for this new class
* Create an all-arg and no-arg constructor, other partial-arg constructors – use constructor chaining to all-arg constructor using this(…)
* Create the “setter” and “getter” methods for all the private instance variables
* Create a “toString()” method
* Create an “equals()” method
* A Test class will be provided to test all methods
* **Inheritance**
* Create a class that extends the class created
* Create one or more instance variables for this new class
* Create a no-arg constructor,

all-arg constructor (that will also contain instance variables from the extended class – use super)

* Create other constructors – use constructor chaining
* Create the “setter” and “getter” methods for all the private instance variables
* Create a “toString()” method
* A Test class will be provided to test all methods
* arrays of primitive data types but arrays of objects are **NOT** included in the test
* static variables and methods are **NOT** in the test.

**Make sure you have completed all learning activities for each session.**

# Revision Activities

1. Create a **Rectangle** class. Create 2 double instance variables called width and length.

* Write two constructors – all arg and no arg. Use 1 for the default values.
* Write the getter and setter for the width and length.
* Write the toString() which will also display the super toString(). Use StringBuilder append method.
* Write a test class for your Rectangle class to test the no-arg and all-arg constructors, getters, setters and toString.

Update the **Rectangle** class:

* Write a method calcArea(). This method should have no parameters passed to it and needs to return the area of the rectangle as a double. The area is (length\*width).
* Write a method calcPerimeter() to calculate the perimeter of a rectangle. This method should have no parameters passed to it and needs to return the perimeter of the rectangle as a double. The perimeter is 2\*(length+width).
* Test both of these methods in your **RectangleTest** class.
* Create an equals method that uses the length and width and test it works for both true and false.

1. Create a **Square** class that extends **Rectangle.** The square class will have no instance variables but will use the Rectangle class to set its length (and width to be the same). Write two constructors – no arg which will call the super constructor with (DEF\_LENGTH, DEF\_LENGTH) and all-arg which will call the super class constructor with (length, length).

Write a test class called **SquareTest** to test the no-arg and all-arg constructors, getter, setter, equals and toString.

**BOOK and PERSON**

1. Define a **Person** class with 3 instance variables called firstName, surname and age. Write 3 constructors – all arg, 2 arg (with firstName and surname) and no arg. Write the getters, setters and toString(). Write a test class for your Person class.
2. Define a **Book** class with 3 instance variables called isbn, numPages and author(Person object).   
   Write the following constructors:   
   Book()  
   Book(String isbn, int numPages)  
   Book(String isbn, int numPages, Person author)  
   public Book(String isbn, int numPages, String firstname, String surname, int age)  
     
   Write the getters, setters and toString().   
   Write the equals method using isbn.   
   Write a test class for your Book class.
3. Create a main class called arrayExercise.

Create an array called numArray to store 10 integers 9,7,2,5,1,2,8,5,9,4

Output each number in the array.

Count how many numbers are 5 or less and output the result.

(There should be 6)

1. Create the following classes

**Address** class with Strings streetNum, streetName, suburb and postcode. Create an all-arg and no-arg constructor, getters, setters, toString and equals (use streetNum, streetName and postcode.

**DevelopedProperty** class that has a double landArea (square metres), String type (eg residential, commercial), String owner, Address address.   
Create 5 constructors:

DevelopedProperty()

DevelopedProperty(double landArea)

DevelopedProperty(double landArea, String type)

DevelopedProperty(double landArea, String type, String owner, Address address)

DevelopedProperty(double landArea, String type, String owner, String streetNumber, String streetName, String suburb,String postcode)

Create getters, setters and toString.

**BlockOfUnits** class that extends DevelopedProperty and has int noOfUnits.

Create 3 constructors:

BlockOfUnits ()

BlockOfUnits (int noOfUnits)

BlockOfUnits (double landArea, String type, String owner, Address address, int noOfUnits)

Create getters, setters and toString.

**Solutions for these exercises are provided in the Self Assessment section.**