

# SET07109 Programming Fundamentals

## Coursework 2

### 1 Overview

This is the second coursework for SET07109 Programming Fundamentals. This coursework is worth **60%** of the coursework total. The coursework is worth **60%** of the module total. Therefore this coursework is worth **36%** of the total marks available for this module.

In this coursework you are to build a class that supports a binary tree data structure for storing `int` data. You will be provided with three files - a header file and supporting C++ file for the binary tree, and a test file to test the code. Your task is to complete the C++ file for the binary file. The binary tree class will require a number of operator overloads to function. For example, to add a number to the binary tree we would use the following line of code:

```
tree = tree + 5
```

The specification will go into detail about how the application should behave and the requirements.

### 2 Specification

The `binary_tree` class is required to store data for fast retrieval in order. This is an extension to the case study covered at the end of Unit 6. The class requires a number of methods to be able to operate as illustrated in Table 1.

All of the printing functions should return a string of the numbers in the tree separated by a space (e.g. "2 3 4 5").

The `binary_tree` class also requires a number of operator overloads. These operators are provided in Table 2.

Method	Description
<code>binary_tree()</code>	Creates an empty binary tree
<code>binary_tree(int value)</code>	Creates a binary tree with an initial value to store
<code>binary_tree(const vector&lt;int&gt; &amp;values)</code>	Creates a binary tree from a collection of existing values
<code>binary_tree(const binary_tree &amp;rhs)</code>	Creates a binary tree by copying an existing binary tree. This must be a deep copy, not a reference.
<code>~binary_tree()</code>	Destroys the binary tree, cleaning up used resources.
<code>void insert(int value)</code>	Adds a value to the binary tree. If the value already exists in the tree nothing happens.
<code>void remove(int value)</code>	Removes a value from the binary tree if it exists.
<code>bool exists(int value)</code>	Returns true if the value is in the tree, false otherwise
<code>string inorder()</code>	Generates a string representing the tree in numerical order
<code>string preorder()</code>	Generates a string representing the tree in <i>pre-order</i> fashion
<code>string postorder()</code>	Generates a string representing the tree in <i>post-order</i> fashion

Table 1: List of Methods for `binary_tree` Class

### 3 Restrictions and Resources

**Your application must be based purely in C++ using the C++ standard library.** Your application must build in a standard compiler such as Microsoft's `cl`, `g++` or `clang++`.

### 4 Submission

**Your code must be submitted to the Moodle Assignment Drop-box by Midnight (12am) on Monday 18th of April.** If your submit late without prior authorisation from your personal tutor your grade will be capped at 40%.

The submission must be your own work. **If it is suspected that your**

Operator	Example	Description
<code>=</code>	<code>tree1 = tree2</code>	Copy assignment operator. Deep copies the tree to another tree.
<code>==</code>	<code>bool res = tree1 == tree2</code>	Checks if two trees are equal. Two trees are equal when they both contain the exact same values.
<code>!=</code>	<code>bool res = tree1 != tree2</code>	Checks if two trees are not equal. Two trees are not equal if they do not both contain the exact same values.
<code>+</code>	<code>tree = tree + 5</code>	Inserts a new value into the binary tree
<code>-</code>	<code>tree = tree - 5</code>	Removes a value from the binary tree
<code>&gt;&gt;</code>	<code>cin &gt;&gt; tree</code>	Reads in values from an input stream into the tree
<code>&lt;&lt;</code>	<code>cout &lt;&lt; tree</code>	Writes the values, <i>in-order</i> , to an output stream

Table 2: List of Operator Overloads for `binary_tree` Class

submission is not your own work then you will be referred to the Academic Conduct Officer for investigation.

All coursework must be demoed. If the coursework is not demoed to a member of the teaching team this will result in a grade of 0. The demo session is the point where the teaching team will give you feedback on your work.

The submission to Moodle must take the form of the following:

- The code file(s) required to build your application
- A collection of test files to test your application on
- A `makefile` to allow building of your application. The `makefile` should build the binary tree as a separate library, and also allow building of the test application against this library.
- A *read me* file indicating how the `makefile` is used and the toolchain used to build it (i.e. Microsoft Compiler, clang, etc.)

These files must be bundled together into a single archive (a zip file) using your matriculation number as a filename. For example, if your matriculation

number is *1234* then your file should be called *1234.zip*. All submissions must be uploaded to Moodle by the time indicated.

## 5 Marking Scheme

The coursework marks will be divided as follows:

Description	Marks
Basic constructors	2
Vector constructor	2
Copy constructor	2
Insert method	2
Remove method	4
Exists method	2
Different printing methods	2
Basic operators (+, -)	2
Assignment operator (=)	2
Equality operators	4
Input / output operators	4
Resources free and cleared up correctly	2
Code quality	2
Makefile works correctly	2
Submission zip folder complete and correctly collated	2
<b>Total</b>	<b>36</b>

To pass this coursework you will require a minimum of 15 marks, although you should be aiming to achieve a much higher mark to illustrate your understanding of the material.

**BE WARNED! This coursework requires an understanding of the material covered in units 6 to 11 of the module. If you do not complete these you will struggle. DO NOT LEAVE THIS WORK TO THE LAST MINUTE!**

If you have any queries please contact a member of the teaching team as a matter of urgency. This coursework is designed to be challenging and will require you to spend time developing the solution.