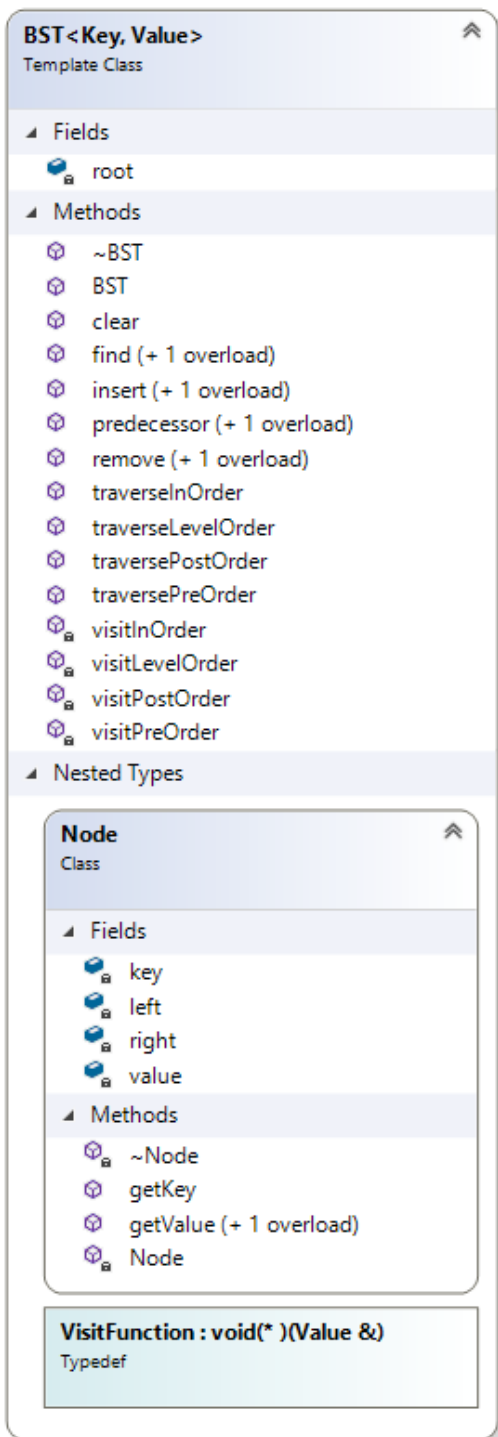


## Assignment Sheet

|  |  |
|--|--|
| <b>Course, instructor name &amp; contact info</b>            | GAME255 – Data Structures and Design Patterns<br>Jean – Paul Amore, jean-paul.amore@humber.ca  |
| <b>Assignment name</b>                                       | Lab 5 – BST Traversal  |
| <b>Grade value</b>   | 6% (Rubrics attached)  |
| <b>Due date</b>  | Week 10  |
| <b>Individual or group assignment</b>                        | Individual. You may consult with your peers on this assignment, but you are not allowed to share your work or submit work that is not yours. Violation of this may result in an academic misconduct penalty. |
| <b>Submission instructions</b>                               | Submit your C++ file BST.h on Blackboard   |
| <b>Targeting these learning outcomes from course outline</b> | <ul style="list-style-type: none"><li>• Be able to use and build vectors, list, stacks, queues, trees and graphs.</li></ul>  |

## Assignment Instructions

Using the Microsoft Visual Studio project provided, complete the functionality for the Binary Search Tree class BST, based on the comments supplied. The BST class contains the following fields and member functions:



## Assignment Instructions

Complete the functionality for the following fields and methods:

- `void visitPreOrder(Node*, VisitFunction)`
- `void visitPostOrder(Node*, VisitFunction)`
- `void visitLevelOrder(Node*, VisitFunction)`

Once you complete the functionality for the BST class, ensure that it executes with the supplied `main()` function.

### SAMPLE OUTPUT

You can review the output by running the executable in the OUTPUT folder.

|  |  |
|--|--|
| <pre>Sorting players by name: Abe: 65 Alice: 95 Bob: 53 Bubba: 18 George: 98 Jim: 23 Joe: 67 Lucy: 33 Sally: 75 Walter: 66  In-Order Traversal Bubba: 18 Jim: 23 Lucy: 33 Bob: 53 Abe: 65 Walter: 66 Joe: 67 Sally: 75 Alice: 95 George: 98  Pre-Order Traversal Jim: 23 Bubba: 18 Sally: 75 Bob: 53</pre> | <pre>Lucy: 33 Walter: 66 Abe: 65 Joe: 67 Alice: 95 George: 98  Post-Order Traversal Bubba: 18 Lucy: 33 Abe: 65 Joe: 67 Walter: 66 Bob: 53 George: 98 Alice: 95 Sally: 75 Jim: 23  Level-Order Traversal Jim: 23 Bubba: 18 Sally: 75 Bob: 53 Alice: 95 Lucy: 33 Walter: 66 George: 98 Abe: 65 Joe: 67</pre> |
|--|--|

**NOTE:** The `main()` function should not be modified for any reason.

## Rubrics

| CRITERIA                             | 0 POINTS  | 1 POINT   | 2 POINTS  | 3 POINTS   | 4 POINTS                                     | 5 POINTS   |
|--------------------------------------|---|---|---|--|--|--|
| <b>1. LOGIC</b>                      | Did not complete assigned work                      | Does not demonstrate ability to use logical process | Poorly demonstrate s ability to use logical process | Somewhat demonstrate s ability to use logical process      | Demonstrate s ability to use logical process | Demonstrate s exceptional ability to use logical process |
| <b>2. EFFICIENCY</b>                 | Did not complete assigned work                      | Does not demonstrate any efficiency                 | Poorly demonstrate s efficiency                     | Demonstrate s some efficiency                              | Demonstrate s efficiency                     | Demonstrate s exceptional efficiency                     |
| <b>3. FUNCTIONALITY</b>              | Did not complete assigned work                      | Barely any code is functional and accurate          | Parts of code are functional, but are not accurate  | Some code is functional and accurate                       | Most code is functional and accurate         | All code is functional and accurate                      |
| <b>4. PROCESS &amp; ORGANIZATION</b> | Did not complete assigned work                      | Very confusing code indentation and/or algorithms   | Somewhat confusing code and/or algorithms           | Satisfactory code and/or algorithms, but could be improved | Good code and/or algorithms                  | Excellent code and/or algorithms                         |
| <b>5. TEST CASE</b>                  | Did not complete assigned work                      | Does not compile                                    | Barely any code functions with test case            | Partially functions with test case                         | Mostly functions with test case              | Fully functions with test case                           |
| <b>6. TIMELINESS</b>                 | Did not submit or submitted five, or more days late | Submitted four days late                            | Submitted three days late                           | Submitted two days late                                    | Submitted one day late                       | Submitted on time  |

## Grading standard

30/30 - Work so amazing the instructor would only see this once in a lifetime

25/30 – Exceptional work, rare

20/30 - Great work, student has full command of the topic.

15/30 - Minor errors

10/30 - Errors and perhaps a major error

5/30 - Regular and consistent major errors. Lack of understanding

2/30 - Largely empty