# Name: An Coklar

Assignment 10: Binary Search Tree

## Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Part** | Metric | Points Earned | Possible Points |
| **Functionality** | 1 | Requirements & stipulations implemented & tested (sunny/rainy day) | 1 | 25 |
| 2 | All canonical functions implemented & tested (sunny/ rainy day) | 1 | 25 |
| 3 | No memory corruption or leaks (including run-time errors) | 10 | 10 |
| 4 | Constant correctness | 1 | 5 |
| 5 | Compiles & links without issues | 0 | 5 |
| **Style** | 1 | Efficiency and simplicity of code | 7.5 | 7.5 |
| 2 | Proper abstraction and code reuse (no duplicated code) | 3.5 | 7.5 |
| 3 | Proper encapsulation & polymorphism, OOP style, proper modifiers for data (public, protected, private), Getters/Setters, no global variables | 6.5 | 7.5 |
| 4 | Code is formatted correctly (no extra white space where it is not needed; program contains indentation where necessary for loops, logic statements, and scope), code is self-documenting or contains comments where appropriate, and code uses a class variable, function naming convention | 7.5 | 7.5 |
|  |  | **Score** | 37 | 100 |

# Functionality

## Part 1: Requirements & stipulations implemented & tested (sunny/rainy day)

Points Score: 1 / 25

|  |
| --- |
| Merits |
| Declared and has implementation of the **BSTree’s** class methods and traversals. Declared and has implementation of the **BSTNode’s** class methods. Implementation of the **BSTNode’s** getters and setters are correct. |

|  |
| --- |
| Demerits |
| (-10) The solution doesn’t contain test for the BSTree class methods.  ~~(-1) BSTree’s Set Visit Function is assigning the parameter visit to the root of the tree. This is causing a compiler error~~ [~~C2679.~~](https://msdn.microsoft.com/en-us/library/h1925w4w.aspx) ~~The compiler error states “binary ‘=’: no operator found which takes a right-hand operator of type ‘BSTNode<K, V>\* (or there is no acceptable conversion). Issue shown below.~~ |

1.

template<

**class**

K,

**class**

V>

2.

**void**

BSTree<K,

V>::SetVisitFunction(

**function**

<

**void**

(

**const**

BSTNode<K,

V>\*

**const**

node)

>

visit)

noexcept

3.

{

4.

visit

=

root\_;

5.

}

(-1) ~~BSTree Delete is unable to match function definition to an existing declaration, causing compiler error~~ [~~C2244.~~](https://msdn.microsoft.com/en-us/library/yf190ysd.aspx) ~~ITree Delete method’s parameter is by value, not by reference. Issue and fix shown below.~~

~~Cause of error:~~

~~1. void Delete(const K & key) noexcept(false) override;~~

~~Fix of error:~~

~~1. void Delete(const K key) noexcept(false) override;~~

**(-1) ~~BSTree’s Clear method is calling Purge, however, Purge hasn’t been defined in the class. Causing compiler error,~~** [**~~C3861,~~**](https://msdn.microsoft.com/en-us/library/b6y87cyx.aspx) **~~“‘Purge’ identifier not found.” Issue shown below.~~**

1.

template

<

**class**

K,

**class**

V>

2.

**void**

BSTree

K,

<

V>::Clear()

noexcept

3.

{

4.

Purge(root\_);

5.

root\_

=

nullptr;

6.

}

**(-1) BSTree’s Height method is calling Height with A BSTNode as an argument, however, this method hasn’t been defined. This cause compiler error,** [**C2660,**](https://msdn.microsoft.com/en-us/library/ek13fhc1.aspx) **the error states “’BSTree<K, V>::Height’: function does not take 1 argument.” A possible fix would be declaring a height method inside the BSTree. Issue shown below.**

1.

template

<

**class**

K,

**class**

V>

2.

size\_t

BSTree

<

K,

V>::Height()

noexcept

3.

{

4.

**return**

Height(root\_);

5.

}

(-7) ~~BSTree’s In Order traversal is attempting to call the root’s in order method, however, this is cause compiler error,~~ [~~C2039,~~](https://msdn.microsoft.com/en-us/library/fdwb3fd7.aspx) ~~the error states “‘InOrder\_’: is not a member of ‘BSTNode<K, V>’”. I think your attention was to call the tree in order method. This error occurs for each of the traversal methods. Issue and fix shown below.~~

~~Cause of error:~~

~~1. root\_->InOrder\_();~~

~~Fix of error:~~

~~1. InOrder\_(root\_);~~

(-1) **~~BSTree’s Is Empty method is attempting to call the root’s ‘isInTree’ method, however, this is causing compiler error,~~** [**~~C2039,~~**](https://msdn.microsoft.com/en-us/library/fdwb3fd7.aspx) **~~the error states “’isInTree’: is not a member of ‘BSTNode<K, V>’. Issue shown below.~~**

**1. root\_->isInTree();**

(-1) ~~BSTree’s Breadth First traversal is using an undeclared identifier, ‘Root’, causing compiler erro~~[~~r C3861.~~](https://msdn.microsoft.com/en-us/library/b6y87cyx.aspx) ~~Issue shown below.~~

~~1. if (Root != nullptr)~~

(-1) ~~The BSTree is attempting to directly use the BSTNode private data members function causing compiler error,~~ [~~C2248.~~](https://msdn.microsoft.com/en-us/library/tsbce2bh.aspx) ~~To resolve these errors, use the node’s getters and setters.~~

~~Cause of error:~~

~~1. root\_->key\_~~

~~Fix of error:~~

~~1. root\_->GetKey()~~

|  |
| --- |
| ~~Suggestions~~ |
| ~~The ITree class methods Insert and Delete are design to be pure virtual functions, and should contain no implementations.~~  ~~Test your implementation. At the bare minimal a call should be made to every class method.~~ |

Part 2: Canonical functions implemented & tested (sunny / rainy day)

Points Score: 1 / 25

|  |
| --- |
| Merits |
| Declared and has implementation of BSTree’s canonical functions.  Declared and has implementation of BSTNode’s canonical functions. |

|  |
| --- |
| Demerits |
| (-10) The solution doesn’t contain any test for the BSTree’s canonical functions.  (-3) ~~The BSTNode’s constructor is attempting to reference PairNode’s default constructor, which is a deleted function causing compiler error~~ [~~C2280.~~](https://docs.microsoft.com/en-us/cpp/error-messages/compiler-errors-1/compiler-error-c2280) ~~The BSTNode’s constructor’s member initialization list needs to call a declared constructor of PairNode. This is also true with BSTNode’s move and copy constructors. Issue and fix shown below.~~ |

~~Cause of error:~~

1.

template

<

**class**

K,

**class**

V>

2.

BSTNode

<

K,

V>::BSTNode(

**const**

K

key,

**const**

V

value,

BSTNode<K,

V>\*

right,

BSTNode<K,

V>

\*

left)

3.

:

left\_(left),

right\_(right)

4.

{

5.

}

~~Fix:~~

1. ~~template <~~**~~class~~** ~~K,~~ **~~class~~** ~~V>~~
2. ~~BSTNode<K, V>::BSTNode(~~**~~const~~** ~~K key,~~ **~~const~~** ~~V value, BSTNode <K, V>\* right, BSTNode<K, V>\* left)~~
3. ~~: left\_(left), right\_(right), PairNode <K, V> (key, value)~~
4. ~~{~~
5. ~~}~~

(-2~~) The BSTNode’s assignment is not copying the data and key from the other node, and the BSTNode’s move assignment is not moving data and key from the other node.~~

(-4) ~~Inside the BSTree copy/move constructors and copy/move assignment is the code below. The variable ‘q’ is being declared as a queue, however, it’s being used as a stack. This is causing compilers error like ‘Pop’ is not a member of ‘ArrayQueue’ and ‘Push’ is not a member of ‘ArrayQueue’. Issue shown below.~~

1.

ArrayQueue

<

BSTNode

<

K,

V>

**const**

\*

>

q;

2.

q.Push(copy.root\_);

3.

BSTNode

K,

<

V>

**const**

\*

ptr

=

nullptr;

4.

**while**

q.IsEmpty

(!

())

{

5.

ptr

=

q.Front();

6.

q.Pop();

7.

**if**

ptr

(

-

>

left\_)

{

8.

q.Push(ptr

-

>

left\_);

9.

}

10.

**if**

ptr

(

-

>

right\_

)

{

11.

q.Push(ptr

-

>

right\_);

12.

}

13.

}

(-5) PairNode’s canonical functions aren’t defined.

|  |
| --- |
| Suggestions |
| Test the implementation of the canonical functions. At the bare minimal constructor each object. |

## Part 3: No memory corruption or leaks (including run-time errors)

Points Score: 10 / 10

|  |
| --- |
| ~~Merits~~ |
| ~~The BSTree’s destructor does delete the root node.~~ |

## ~~Part 4: Constant Correctness~~

~~Points Score: 1 / 5~~

|  |
| --- |
| ~~Demerits~~ |
| (-4) ~~The BSTree’s private traversal methods should have constant qualifier. Issue and fix shown below.~~ |

~~Issue:~~

1. **~~void~~** ~~InOrder\_(~~**~~const~~** ~~BSTNode<K, V>\*~~ **~~const~~** ~~node);~~
2. **~~void~~** ~~PreOrder\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node);~~
3. **~~void~~** ~~PostOrder\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node);~~
4. **~~void~~** ~~BreadthFirst\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node);~~

~~Fix:~~

1. **~~void~~** ~~InOrder\_(~~**~~const~~** ~~BSTNode<K, V>\*~~ **~~const~~** ~~node)~~ **~~const~~**~~;~~
2. **~~void~~** ~~PreOrder\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node)~~ **~~const~~**~~;~~
3. **~~void~~** ~~PostOrder\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node)~~ **~~const~~**~~;~~
4. **~~void~~** ~~BreadthFirst\_(~~**~~const~~** ~~BSTNode <K, V>\*~~ **~~const~~** ~~node)~~ **~~const~~**~~;~~

## Part 5 Compilers & links without issues

Points Score: 0 / 5

|  |
| --- |
| Demerits |
| (-5) The project doesn’t compiler without any issues. Read this document in its entirety to see the issues. |

|  |
| --- |
| Suggestion |
| Implement test on all methods and canonicals, the compiler will do a pass over and not detected errors. |

# Style

Part 1: Efficiency and simplicity of code.

Points Score: 7.5 / 7.5

|  |
| --- |
| Merits |
| The code is broken up into small functions. |

|  |
| --- |
| Suggestions |
| Getters and setters that are a single line of code, can be defined in the class declaration to save space. |

## Part 2: Proper abstraction & code reuse (no duplicated code)

Points Score: 3.5 / 7.5

|  |
| --- |
| Demerits |
| **~~(-2) The BSTree copy constructor and assignment operator are identical.~~**  **~~(-2) The BSTree move constructor and move assignment operator are identical.~~** |

## Part 3: Proper encapsulation & polymorphism, OOP style, proper modifiers for data (public, protected, private), Getters/Setters, no global variables

Points Score: 6.5 / 7.5

|  |
| --- |
| Merits |
| No global variables. |

|  |
| --- |
| Demerits |
| (-1) ~~The BSTree’s root data member should be private.~~ THIS WASN’T PRIVATE TO BEGIN WITH |

## Part 4: Code is formatted correctly (no extra white space where it is not needed; program contains indentation where necessary for loops, logic statements, and scope), code is self-documenting or contains comments where appropriate, and code uses a class variable, function naming convention

Points Score: 7.5 / 7.5

|  |
| --- |
| Merits |
| Code is formatted correctly. |

|  |
| --- |
| Suggestions |
| Inside BSTree’s breadth first traversal you declare a variable ‘TempNode.’ It’s conventon for variables declare inside functions to be camel case.  Find ways to take advantage of method overloading instead of renaming traversal methods. |