

### **Pdf.cc functions**

PDFStream::append() : concatenates two strings  
PDFStream::init() : create the initial size of the stream  
PDFStream::grow() : increase the size of the stream  
PDF::init() : initializes the pdf  
PDF::init\_page() : initializes the pdf page  
PDF::finish\_page() : writes in the annotation at the top of the page  
PDF::new\_page() : creates a new page in the pdf  
PDF::destroy() : deletes the pdf  
PDF::selectfont() : selects the font for the pdf  
PDF::arc() : for drawing arcs  
PDF::arcn() : for drawing arcs  
PDF::rectpath() : for drawing rectangles  
PDF::round\_box\_path() : for drawing round boxes  
count\_lines() : line count  
stringwidth() : adjusts how wide a string is on a page  
stringwidth\_multiline() : adjusts how wide multiple strings are on a page  
PDF::position\_text() : for positioning text on a page  
PDF::text\_box() : for creating a text box on a page  
PDF::finish() : finalizes and outputs the pdf

### **PDF.h functions**

font\_index() : a fix for symbol and zapfdingbats fonts  
PDFColor() : constructor for making color  
PDFStream() : constructor for making a pdf stream  
PDFStream::destroy() : deletes text  
PDFPage() : constructor for making a page  
PDFPage::destroy() : deletes page  
PDF() : constructor for making a pdf  
get\_width() : gets the width of the page  
get\_height() : gets the height of the page  
closepath\_fill\_stroke() : operator used for drawing  
closepath\_eofill\_stroke() : operator used for drawing  
eofill\_stroke() : operator used for drawing  
curveto() : operator used for drawing  
fill() : operator used for drawing  
eofill() : operator used for drawing  
setgray() : operator used for drawing  
setgray\_stroke() : operator used for drawing  
setgray\_nonstroke() : operator used for drawing  
closepath() : operator used for drawing  
setflat() : operator used for drawing  
setlinejoin() : operator used for drawing  
setlinecap() : operator used for drawing

setcolor() : operator used for drawing  
setcmykcolor\_stroke() : operator used for drawing  
setcmykcolor\_nonstroke() : operator used for drawing  
lineto() : operator used for drawing  
moveto() : operator used for drawing  
setmiterlimit() : operator used for drawing  
endpath() : operator used for drawing  
gsave() : operator used for drawing  
grestore() : operator used for drawing  
setcolor() : operator used for drawing  
setcolor\_stroke() : operator used for drawing  
setcolor\_nonstroke() : operator used for drawing  
setrgbcolor() : operator used for drawing  
setrgbcolor\_stroke() : operator used for drawing  
setrgbcolor\_nonstroke() : operator used for drawing  
closepath\_stroke() : operator used for drawing  
stroke() : operator used for drawing  
shfill() : operator used for drawing  
setlinewidth() : operator used for drawing  
clip() : operator used for drawing  
eoclip() : operator used for drawing  
begin\_text() : begins writing text  
end\_text() : stops writing text  
next\_line() : moves text to the next line  
next\_line\_leading() : moves text to the next line while setting spacing  
text\_matrix() : makes a text matrix  
T\_star() : moves to the start of the next line  
show() : shows a string of text  
show\_next\_line() : moves to the next line and shows the string  
show\_next\_line() : moves to the next line and shows the string and sets spacing

## **BinaryTree.h**

BTNode() : constructor for a binary tree node

BinaryTree() : initializes the binary tree

is\_empty() : checks if tree is empty

height() : checks the height of the tree

node\_count() : returns the number of nodes

leaf\_count() : returns the number of leaves

empty\_this() : empties a tree if tree elements are different

init\_complete() : create a new node with left and right children

to\_flat\_array() : copies elements of a tree into an array in complete tree order

preorder() : preorder traversal of the tree

inorder() : places the nodes from left to right

postorder() : recursively goes through all the left tree nodes then the right tree nodes

display() : displays the tree

insert\_node() : inserts a node

delete\_last\_node() : deletes the last node of the tree

delete\_node() : deletes a node from the tree

get\_root() : returns the root node

compare() : compares two nodes to see if they're equal

clone() : clones a node

## **BinaryTree.cpp**

operator<<() : writes elements from the nodes of tree in inorder traversal

push() : pushing nodes to the queue

front() : traversing to the front of the queue

pop() : removing from the queue

**Main.cc**

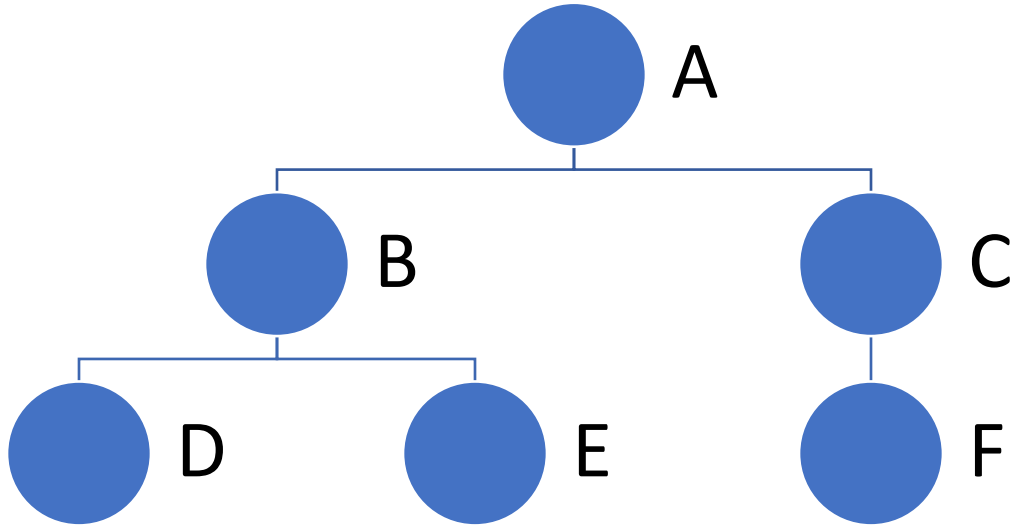
func2() : takes the address of the int given and can modify it elsewhere

complete\_tree\_height2() : returns the height of the complete binary tree

main() : main function that starts the program

### In-Order Traversal

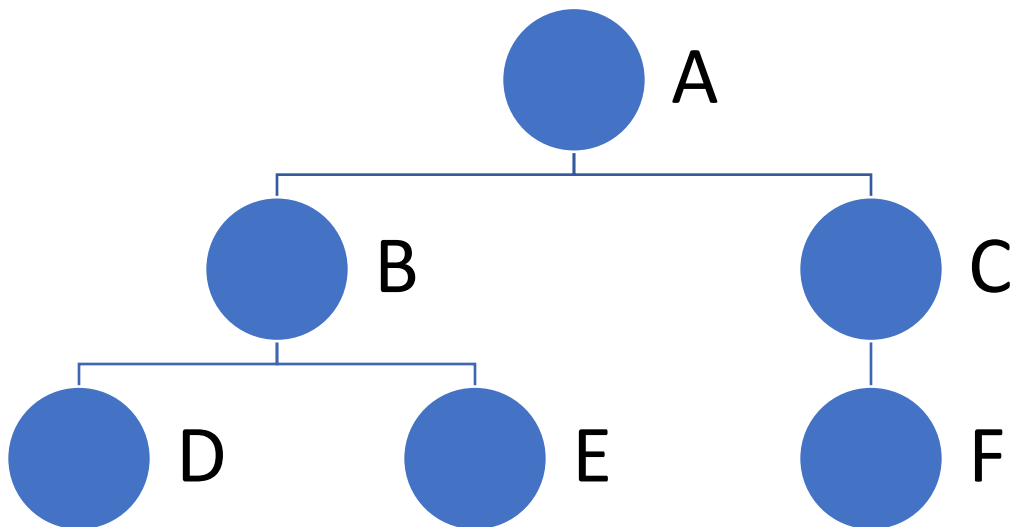
In order traversal is performed by accessing left the subtree of a node, then the parent node itself, then the right subtree of the node.



In order: D B E A F C

### Post-Order Traversal

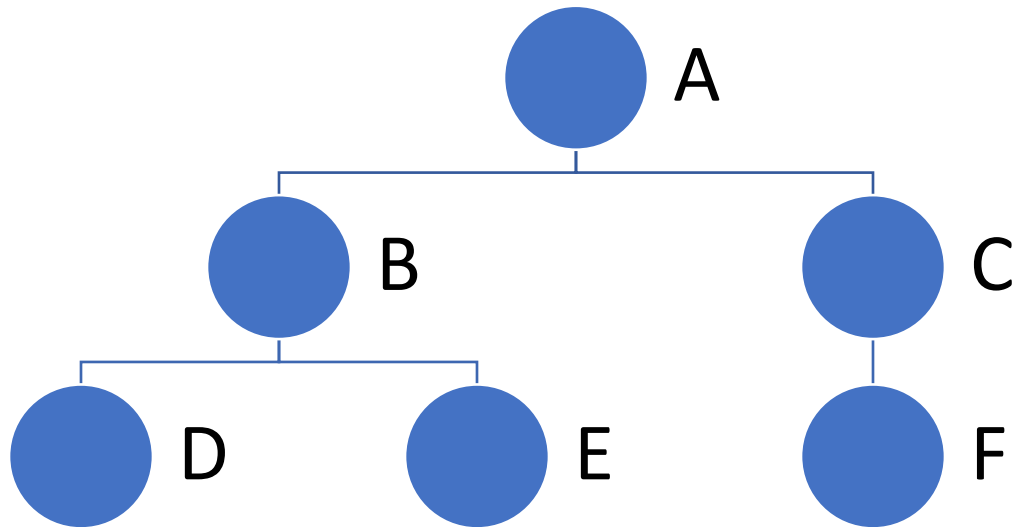
Post order traversal is performed by accessing the left subtree of a node, then the right subtree of a node, then the parent node itself.



Post order: D E B F C A

### Pre-Order Traversal

Pre order traversal is performed by accessing a node, then accessing its children from left to right.



Pre order: A B D E C F