COMP2521 24T1
Assignment 1

Kevin Luxa
cs2521@cse.unsw.edu.au

https://cgi.cse.unsw.edu.au/~cs2521/24T1/assignments/ass1/
• Worth 15% of your final mark
• Due 8pm Monday of Week 7 (25 March)
• Late penalty is 0.2% deducted from your attained mark per hour late
Goal: implement a Set ADT using a binary search tree

- Part 1: Basic operations
  - insert, delete, contains, size, print
- Part 2: Common set operations
  - union, intersection, equals, subset
- Part 3: Using a balanced BST
- Part 4: Index operations
  - get element at index, get index of element
- Part 5: Cursor operations
  - use a cursor to move between elements of a set
The concrete representation is defined in SetStructs.h.

```c
struct set {
    struct node *tree;
};

struct node {
    int item;
    struct node *left;
    struct node *right;
};
```

You must use these fields as described in the spec. You will need to add extra fields to these structs in most parts of the assignment.
Data Structures

struct set

struct node

item 5

left

right

item 2

left NULL

right NULL

item 8

left NULL

right NULL

Data Structures

struct set

tree

item 5

left

right

item 2

left NULL

right NULL

item 8

left NULL

right NULL
testSet.c contains assert-based tests.

- **assert** takes an expression (usually a condition) and crashes the program if the expression evaluates to 0 (or false)
- Each test creates one or more sets, performs some operations on the sets, and then checks that some condition is true using **assert**
- For example:

```c
Set s = SetNew();
SetInsert(s, 5);
assert(SetSize(s) == 1);
```

- The current tests are very simple. You will need to add your own tests to properly verify that your code works.
RTFS
Read The Spec
Constraints

The spec lists all the constraints you have to follow. If your approach doesn’t violate the constraints, then it is allowed.

Assumptions

The spec lists all the assumptions you can make. If the spec doesn’t say you can assume something, then you can’t assume it.