

SCC120 Week 7 workshop

Set ADT and Stack ADT

1) Describe the concept of abstraction and how it applies to an abstract data type (ADT).

2) Complete the following definition of a class for the Set ADT (similar to the example for Stack in lecture). There should be at least four methods for the Set ADT (see the lecture slides).

```
public class Set
{
    public Set();
    ...
}
```

3) For the Set ADT using a linear array, we wrote the *size()* method as follows.

```
int noElements = 0;
int i = 0;
while ((i < limit) && (S[i] != null)) {
    noElements++;
    i++;
}
return noElements;
```

We can make a simple change to make this code shorter while computing the same information. How can we do this?

4) For the Set ADT using a linked list, write pseudocode for the *exists(X)* and *add(X)* methods. The *exists(X)* method finds whether or not element *X* is in the set. What is their worst-case $O()$? Explain why (using your code).

5) Give a real-life example where the Set ADT is useful. And give a real-life example where the Stack ADT is useful.

6) For the Stack ADT, we went through in lecture how to implement it with a linear array and with a linked list, for both the *push* and *pop* operations. Go through the code in the lecture slides again by yourself to understand what they do.

7) How can you use a sequence of stack operations (i.e. *push* and *pop*) to reverse a list of numbers. Describe your answer in words or pseudo code.

8) For the Stack ADT, how can you use only the *push* and *pop* operations to write a function to find the largest item in a given stack? Describe your answer in words.