C++ `Student` class [problem 1]

```
Student(const String &name);
                      // create a student with the specified name and a total score of 0
void addScore(int score);      // adds score to the total score for this student
int getScore() const;               // returns the total score for this student
string getName() const;             // returns the name of this student
```

Java `String` class (selected methods)

`char charAt(int index)`
: Gets the char at position index (counting from 0)

`int length()`
: The number of characters in the string.

`String substring(int beginIndex, int endIndex)`
: Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`.

`String substring(int beginIndex)`
: Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the end of this string.

Java `Stack<ElmtType>` class

`Stack()`
: Creates an empty stack.

`boolean empty()`
: Returns true iff the stack is empty.

`ElmtType peek()`
: Looks at the object at the top of this stack without removing it from the stack.

`ElmtType pop()`
: Removes the object at the top of this stack and returns that object as the value of this function.

`void push(ElmtType e)`
: Pushes an item onto the top of this stack.

C++ `Node` type and `ListType` [problem 6]

```
struct Node {
  int data;
  Node * next;
  Node() { data = 0; next = NULL; }
  Node(int d) { data = d; next = NULL; }
  Node(int d, Node * n) { data = d; next = n; }
};

typedef Node * ListType;
```