Question 1

object-oriented programming :: general knowledge	

Give brief definitions of the following terms:

- object
  A collection of data (what the object is made of) and methods (or member functions) on that data (what the object can do, or what messages the object understands) having a type (what the object is seen as).

- class
  A description of (or a template (though not in the C++ and Java sense) for creating) an object.

- instance
  A specific object existing in memory.

- superclass
  A class that is a parent of the class to which we are referring (i.e. a class from which we directly or indirectly inherit).

- subclass
  A class that is a child of the class to which we are referring (i.e. a class that directly or indirectly inherits from us).

- member
  A datum or method belonging to a class.

- method
  Or member function. An action that an object is able to perform, or a message that an object is able to understand.

- getters and setters
  Methods used for retrieving (getting) or modifying (setting) an object's data members.

- accessors and mutators
  The same as getters and setters.

- access modifiers
  - public
    Visible to everyone.
  - private
    Visible within the class, and to friends.
  - protected
    Visible within the class, to child classes, and to friends.

Question 2

classes	
documentation	
consistency of style	

Preface your program with a block comment containing a short description of what it does, and with all necessary `#include` and `using` statements.

Next, write a `Character` class with the following members:

- A `name`, as a string, with either `private` or `protected` visibility.
- A default constructor that optionally accepts a constant reference to a `string` and initializes the `name` data member to that value.
- A `void` method named `sayName` that prints `"My name is "` followed by the `Character`'s name, followed by a newline, to standard output.

Next, write an `AvatarCharacter` class that inherits from `Character` and has the following members:

- An `element`, as a string, with either `private` or `protected` visibility.
- A default constructor that optionally accepts two constant references to `string`s, passes the first value to the parent constructor, and uses the second to initialize the `element` data member.
- A `void` function named `sayElement` that sends `"I bend "`, followed by the `AvatarCharacter`'s `element`, followed by a newline, to standard output.

Finally, write a short `main()` that calls the `sayName()` and `sayElement()` methods on objects of type `AvatarCharacter` to produce the following output:

Output

```
My name is Aang
I bend everything!
My name is Katara
I bend water :)
My name is Sokka
I bend nothing :(
```

Before you begin, please

- Be sure to write your answers neatly and in a good and consistent style. This will be graded. I recommend writing out your solution on the back of the test or on scratch paper and then copying it to page 3 (make sure to cross out the version you don't want graded). You might also consider using pencil, if pen is your usual choice.
- Try to comment with the level of detail you would find helpful (but not irritating) in code of similar complexity written by another student in this class.

```
/**
 * Short program to make 3 characters from Avatar say their name and what
 * element(s) they bend.
 */

#include <iostream>
using std::cout;
using std::endl;

#include <string>
using std::string;

// ...........................................................................

class Character {
    protected:
        string name;

    public:
        Character(const string & name = "") : name(name) {}

        void sayName() { cout << "My name is " << name << endl; }
};

class AvatarCharacter : public Character {
    protected:
        string element;

    public:
        AvatarCharacter(const string & name = "", const string & element = "")
            : Character(name), element(element) {}

        void sayElement() { cout << "I bend " << element << endl; }
};

// ...........................................................................

int main() {
    AvatarCharacter characters[] = {
        AvatarCharacter("Aang", "everything!"),
        AvatarCharacter("Katara", "water :)"),
        AvatarCharacter("Sokka", "nothing :("),
    };

    for (int i = 0; i < 3; i++) {
        characters[i].sayName();
        characters[i].sayElement();
    }

    return 0;  // success
}
```