Have you seen anything coll and CS related in the news?
new is not Abstract

- The GoF book presents a whole set of creational patterns for the simple reason that new doesn't allow abstraction.
- When you call “new Type...” the result is of type Type. Always.
- This can be very limiting.
- It leads to a number of different design patterns aimed at abstracting the object creation process.
You have multiple instantiation points that are supposed to create object from a “family”.

The abstract factory interface has methods for making the different objects that are needed.

Concrete implementations instantiate objects of the correct type.

Makes it very easy to add new, different families.
Adding Instantiated Types

- This is the main liability of the abstract factory.
- Your top level is basically an interface. Adding a new type of thing to the family is adding a method to that interface. Requires alteration of all subtypes.
Pattern Fragility

- Patterns are great when properly implemented.
- There are many things you can do wrong.
General Rules

- Program to interface, not implementation.
- Bind to implementations at the right places.
- Use a pattern consistently.
- Refactor intelligently so code ages well.
Closing Remarks

- Groups.
- Have a good weekend.
- Is anyone considering the IEEE robotics competition?