CS 211 - 11/28/17

Tuesday, November 28, 2017 4:53 PM

Exam 2 Next Week during Lab - No exam during final exam week

No lecture next Tuesday.

Exam 2 will be similar to Exam 1.

Write a C++ class that will implement some methods

- Could be a Stack, Queue, or List
- Could be implemented as a Linked List or a Dynamic Array

Possible methods for a List class

- insert/add (may need to be stored in increasing order in the list)
- remove/delete
- getNthItem
- list Length
- does Value X exists

Exam 2 will be closer to Project 6 than Project 7

You will (hopefully) have access to the internet
You have access to any existing code
You have access to any books or notes
You may use your own laptop or a CS machine

Maybe will include the use of makefiles and multiple source code files

1. 1 file for the CLaSSES header file
2. 1 file for the Class methods (includes the .h file)
3. 1 file for main and other non-class code (includes the .h file)
4. 1 makefile

Expect to need multiple instances of the class in the main section of code
perhaps an array of instance (If I am feeling very evil)

==========

Project 7

```
class Creature
{
  private:
       int type;

 public:
      Creature()
      {
         setType (0);
      }

       void setType (int t)
       {
          type = t;
       }

       int getType ()
       {
          return type;
       }

       virtual void dailyActions()
```

```
        {
        }
};

class Ant : public Creature
{
    Ant()
      {
         setType (1);
      }

    void dailyActions() override       // keyword define in C++11
    {
      move();
      spawn(3);
     }
};

class DoodleBug : public Creature
{
    DoodleBug()
      {
         setType (2);
      }

    void dailyActions ( )
    {
      bool successful = hunt();
      if ( !successful )
         move();
```

```
      spawn(8);
      starve();
    }
};

void doMove(Creature* c)
{
    c->move();          // calls move() from the class for the actual parameter
}                       //   type used at the call to doMove()

int main (int argc, char** argv)
{
  Creature* c = new Creature();
  c->move();             // calls move() defined in Creature()
  doMove( c );

  c = new Ant();
  c->move();             // calls move() defined in Ant()
  doMove( c );

  c = new DoobleBug();
  c->move();             //  calls move() defined in DoodleBug()
  doMove ( c );
}
```