

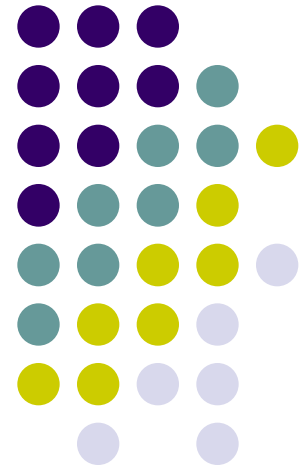
CS 460

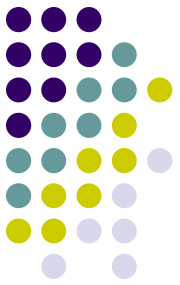
Programming Languages

Fall 2021

Dr. Watts

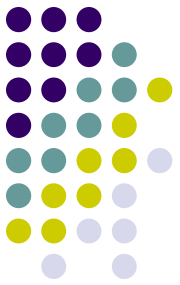
(23 October 2023)





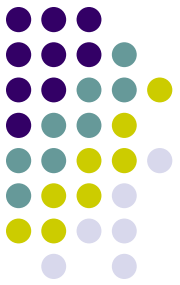
Class money

- money.h
- No changes to public section of class
- size, currency required



Exercise 2

- Your secret folder:
 - ~tiawatts/public_html/cs460f23/Exercise2/abcxyz
- You can copy this folder to your account and modify it.
- Testing groups
 - Identify issues in your own class
 - Design a testing framework for Exercise 5
 - Not code sharing groups

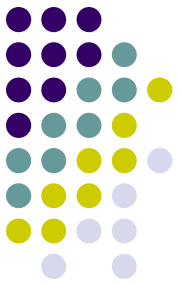


Exercise 5

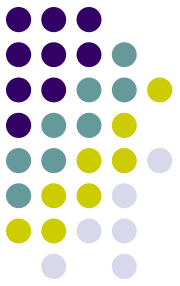
- Will be posted around Thanksgiving
- Complete class
- Doxygen website
- .h vs .cpp documentation
 - .h – for the user
 - .cpp – for the (modifying) programmer
- Application program

Classes

- Constructors
- Destructors
- Accessors
- Mutators



Class money

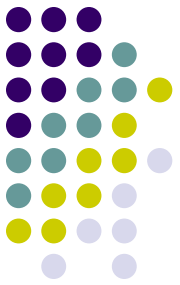


```
private:
```

```
// Add attributes private member functions here.
```

```
    bool positive;  
    unsigned dollars;  
    unsigned cents;
```

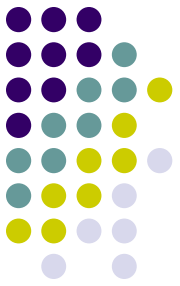
```
    unsigned size;           // required  
    unsigned * currency;    // required
```



Classes

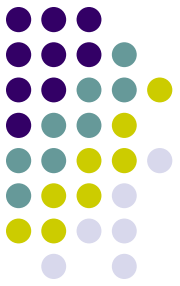
- Operator overloading
 - A bit controversial
 - C++ yes – Java no...
- Assignment operator
 - What if you don't overload it?
 - Dynamic memory issues

Assignment operator for money class

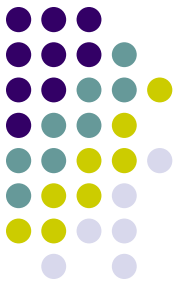


- `money & operator = (const money & other);`
- Check for self assignment
- Deallocation and reallocation of dynamic memory
- Copy
- Return `* this`
- What is “this”?

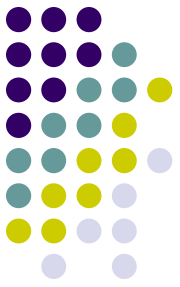
Arithmetic operator overloading



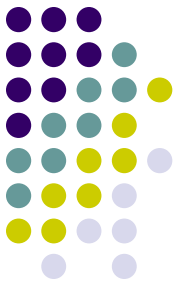
++ and -- operators

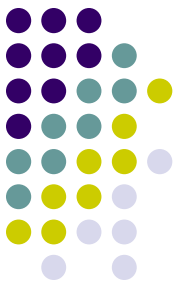


<< and >> operators



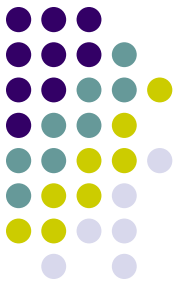
Project 2





What is Syntactic Analysis?

- Also known as Parsing
- Determining if the order of the tokens generated for the lexemes of the input are in a legal order according to some grammar
- Creation of a parse tree
 - Explicit or implicit
- Error recovery
 - When an error is detected, the parser must get back to a normal state and continue analysis of the input
- Basis for translation



VS Semantic Analysis

- The meaning of the expressions, statements and program units.
- Static semantics
 - At compile time
- Dynamic semantics
 - At run time
- Attribute grammars
- Denotational semantics