CSCI251/CSCI851 Autumn-2020
Advanced Programming (**LT7**)

Lecture Tutorial 7

From the lab:

- Constructor initialization list.
  - Fixing Hat … → default constructor or fixing the constructor initialization lists, the latter is tidier.

```
Person::Person(string _name, string
_idNum, string hat_type, char
hat_size): name(_name),
idNum(_idNum),
myHat(hat_type,hat_size){};
```

- Casting:
  - What is casting?
    - Converting one type to another.
- We sometimes see explicit casting …

```
int intValue = 4;
float floatValue = (float) 4;
```

- … but it can also be implicit, automatically used:
- float floatValue = intValue;

- The use of the word implicit in the original lab version was misleading sorry, it was changed to providing a “form of casting”.
- A constructor like X(int) is taking an int and producing an X object.
  - In the sense it’s converting an integer to an X, like casting might.

Better π …

- There are quite a few things that can be done.
  - Variables don't need to have be local and re-declared each time.
    - Why reset the maximums each time?
  - Square roots are expensive.
    - But you are checking if the distance is less than or equal to 1.
    - But if a value is greater than 1 then the square root of that value will be greater than 1.
    - And if a value is between 0 and 1 then the square root that value will between 0 and 1 too.
    - So we don't need the square root at all.

- We don’t need to calculate dPi each time, and we don’t need dCArea, or dI.
- More efficient randomness with ranging likely eliminating the need
- It’s still not going to be an efficient method of calculating π, but the implementation can certainly be significant improved.

- Relationships : This is relevant to extending Goldilocks, and to assignment 2.

No.	Scenario	Relationship	Notes
1	Library, books.	Aggregation	Libraries have many books but the books and library can exist and function fully without specific instances of each.
2	Insects, legs, wings.	Composition	Insects aren't going to fully function with the legs and wings they will typically have.
3	Pet shop, big dogs, little dogs.	Aggregation (shop to dog) and instantiation (dogs)	Pet shops can have dogs. Specific pet shops and dogs can exist without each other. Big dogs and little dogs are instances of the same class, and are unlikely to be different classes.

No.	Scenario	Relationship	Notes
4	Insects, ants, spiders.	Inheritance. But not with insects as a base class for ants and spiders.	Spiders are not insects. Ants are a special case of insect so an Ant class could be derived from an Insect class. Insect and Spider would have a common ancestor (Bug?)
5	Pencil cases, pencils.	Aggregation.	Pencil cases can have many pencils but the cases and pencils can exist and function fully without specific instances of each.
6	Robots, androids, humans.	Multiple inheritance.	Androids are humanoid robots. So androids are derived from robot and humanoid, a base class for the human class representing the shape.