CSCI 1370

May 1, 2017

CLASSES:
RECAP & CONTINUATION

CLASSES

- In C++, **classes** provide the functionality necessary to do *object-oriented programming* (OOP).

- OOP is a way of organizing programs.
  - It doesn't allow anything we couldn't already do, but rather, improves **readability** and **efficiency**.

```
class Student
{
   public:
        string name;
        void setID(double num);
        double getID();

   private:
        double id;
};

void Student::setID(double num)
{
   id = num;
}
```

```
#include “Student.h”


int main()
{
  Student s;
  s.name = “Cynthia”;
}
```

```
class Date
{
  int day, month, year;
  void print();
};

void Date::print()
{
  cout << month << “/” << day << “/” << year;
}
```

CLASS CONSTRUCTORS

- Right now, if we were to create a Date object:

```
Date d;
```

- The member fields could be filled with nonsensical data...

CLASS CONSTRUCTORS

- When an object is declared, e.g.:

```
Date d;
```

  - Memory is allocated.
  - The class' constructor is called.

```
class Date
{
   int day, month, year;
   void print();
   Date();
};

Date::Date()
{
   day = 0;
   month = 0;
   year = 0;
}
```

CONSTRUCTORS W/PARAMETERS

- What if we want to set the initial values to something else?
- We can define *additional* constructors.

```
class Date
{
   int day, month, year;
   void print();
   Date();
   Date(int d, int m, int y);
};

Date::Date()
{
   day = 0;
   month = 0;
   year = 0;
}

Date::Date(int d, int m, int y)
{
   day = d;
   month = m;
   year = y;
}
```

CONSTRUCTORS W/PARAMETERS

- Now we have options when we declare new Date objects:

- `Date someDay;`
- `Date today(5, 1, 2017);`

EXERCISE: PIZZA CLASS

- With a partner, write a class definition for *pizza*.
- The following data should be associated with it:
  - number of toppings
  - cheese level (-1 = low, 0 = moderate, 1 = high)
  - price
- There should also be the following methods:
  - void setToppings (int i): sets the number of toppings
  - void changeCheesiness (bool b): adjusts the cheese level up or down by one
  - void printPrice(): prints out the price.

EXERCISE: PIZZA CLASS

- Add a default constructor for the *pizza* class such that the member fields are set as follows:
  - number of toppings = 0
  - cheese level = 0
  - price = 5.00;
- Add an additional constructor that allows the member fields to be set to the following:
  - number of toppings = input parameter
  - cheese level = input parameter
  - price = computed as $5 + $1*number of toppings + $1 if cheese level not 0

EXERCISE: PIZZA CLASS

- Now, in `main`, declare three different pizzas:
  - declare one (with the default settings)
  - declare one with two toppings and extra cheese
  - declare one with the default settings and then update the cheese level to low

WORKSHOP

CHALLENGE: programming Pokemon Go

- Imagine Pokemon Go has not yet been released and you are tasked with programming (a reduced version) of the game.

- Requires:
  - a **pokedex**
  - 5 **pokemon** (bulbasaur, caterpie, jigglypuff, pidgey, pikachu)
  - each pokemon must have:
    - a **name**, **weight**, and **strength**, and
    - the ability to **evolve** and ability to **power up** (increase strength)

CHALLENGE: programming Pokemon Go

- Find a partner.
- Discuss – at a conceptual level – your ideas for writing this program using OOP.
- Check-in with me when ready to continue to the next item.

- Requirements:
  - a **pokedex**
  - 5 **pokemon** (bulbasaur, caterpie, jigglypuff, pidgey, pikachu)
  - each pokemon must have:
    - a **name**, **weight**, and **strength**,
    - the ability to **evolve** and ability to **power up**

DISCUSSION

```
class Pokemon {
    boolean isEvolved;
    double strength, weight;
    string name;
    void evolve(string s);
    void powerUp(double num);
};

void Pokemon::evolve(string s)
{
    if (!isEvolved)
    {
            name = s;
            weight += 50;
            isEvolved = true;
    }
}

void Pokemon::powerUp(double num)
{
    strength += num;
     weight += num;
}
```

```
using namespace std;

int main()
{
   Pokemon bulbasaur;
}
```