Unit 11.2 Highlights

A node module encapsulates related code into a single unit of code.

When writing a node module, before anything is exported, the value of `module.exports is {}` (an empty object). The value of `module.exports` can be replaced or appended with a number, boolean, string, or another object (function, object literals, expression, arrays, etc.). Adding / updating the value of `module.exports` allows importing applications an opportunity to use methods, constructors, etc. the module has made publicly available through `module.exports`

Ex. 1 (exporting constructors within node modules)

```
//team.js
function Team(members, nickname) {
    this.members = members;
    this.nickname = nickname;
}
module.exports = Team; //making the Team constructor publicly accessible

//------------------------------ meanwhile, in a separate file called member...

//member.js
function Member(name, city, skill) {
    this.name = name;
    this.city = city;
    this.skill = skill;
}
module.exports = Member; //making the Member constructor publicly accessible
```

Ex. 2 (using constructors exported from node modules to create instances of Team and Member)

```
//keyword require returns an object, which references the value of module.exports for the selected file
var Team = require('./team.js');
var Member = require('./member.js');

//creating new instances of the Member object
var bs = new Member('Bryce Swackhammer','Raleigh', 'Hit players running w/ football')
var lb = new Member('Legend Brumbaugh','Ann Arbor','Running w/ football');
var tw = new Member('Trent Wilderbraithwaite','Huntington','Hit players running w/ football');
var gg = new Member('George Georgopoulos','Amherst','Kick football');

//creating a new instance of the Team object
var squad = new Team([bs,lb,tw,gg],'French Toast Mafia');
```