Node Modules: Callbacks and Error Handling

Jogesh K. Muppala

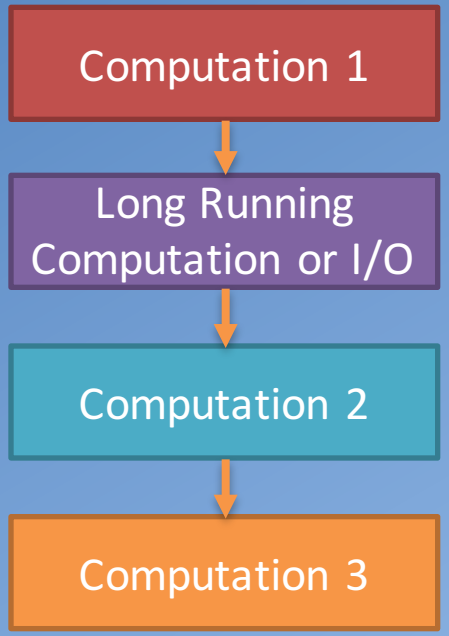
THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

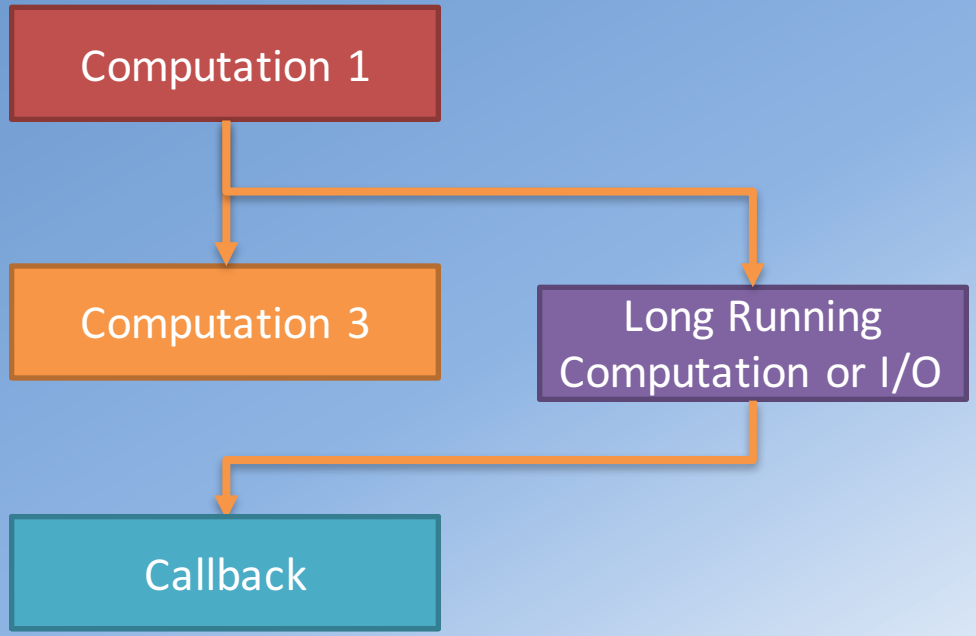
Two Salient Features of JavaScript

- First-class functions: A function can be treated the same way as any other variable
- Closures:
  - A function defined inside another function has access to all the variables declared in the outer function (outer scope)
  - The inner function will continue to have access to the variables from the outer scope even after the outer function has returned

Asynchronous Programming

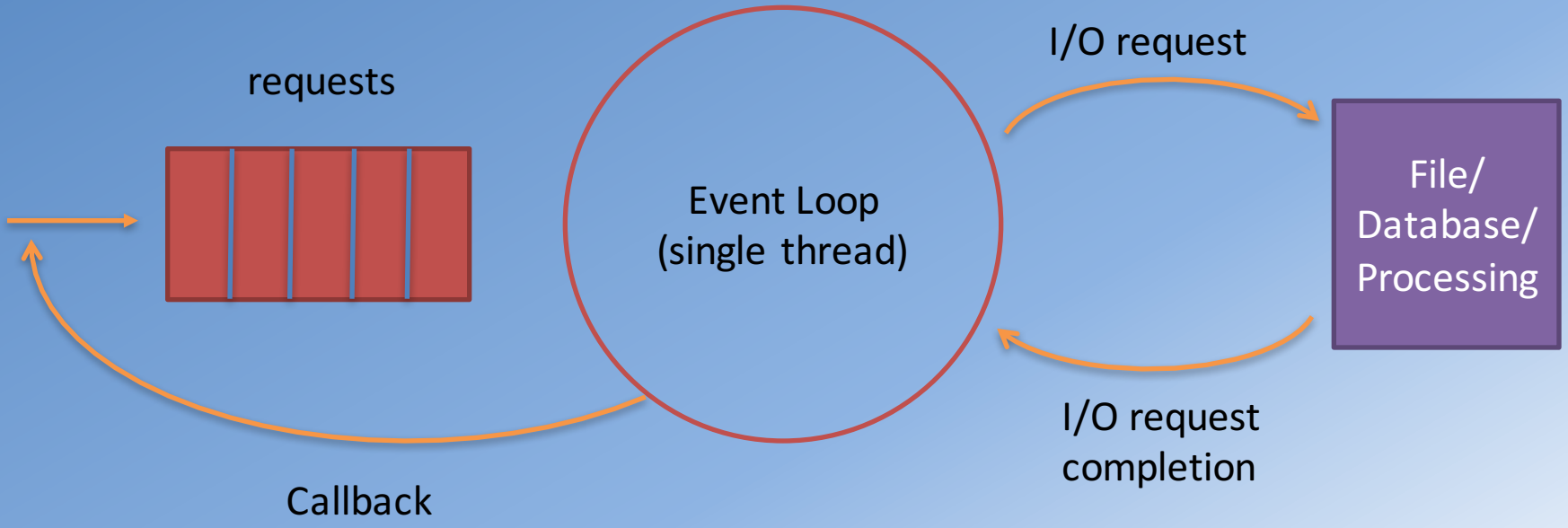


Synchronous Programming



Asynchronous Programming

Node, Async I/O and Callbacks



Callbacks and Error Handling

- rectangle module:

```
module.exports = function(x,y,callback) {
 try {
  if (x < 0 || y < 0) {
     throw new Error("Rectangle dimensions should be greater than zero: l = " + x + ", and b = " + y);
  }
  else
     callback(null, {
        perimeter: function () { return (2*(x+y)); },
        area:function () { return (x*y); }
     });
 }
 catch (error) { callback(error,null); }
}
```

Callbacks and Error Handling

- Calling the function:

```
rect(l,b, function(err,rectangle) {
        if (err) {
                console.log(err);
        }
        else {
                . . .
        }
});
```

Exercise: Node Modules: Callbacks and Error Handling

- Using Callbacks and error handling in Node applications
- Using external Node modules