

CS Client-Server Programming
Spring Semester, 2006
Doc 1 Intro
Jan 19, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Course Web Site

<http://www.eli.sdsu.edu/index.html>

on-line courses

CS 580 Spring 06

Lectures with Ink

<http://up.ucsd.edu/>

Sign in

Create your own account

<http://up.ucsd.edu/classrooms/SDSUCS580/>

Sign in using class account

In class notations

Input from students on-line

Languages

Java

Smalltalk

Ruby

C#

Knowing a Language

Basic syntax of the language

Core API

No one knows the entire API of either language

You should have good grasp of the common or core API

Collections, Files, Exceptions, Streams

Language culture - Ways of doing things in each language

Java Doc

Searching the API

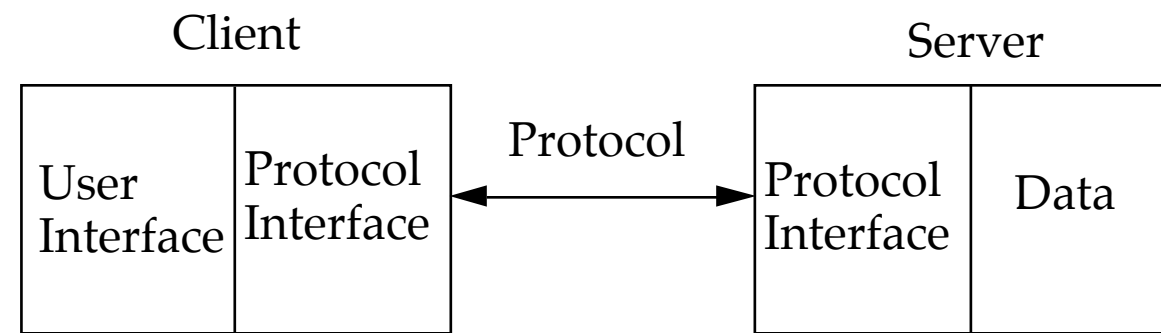
Compiling/running code

Using Smalltalk browsers

Naming conventions

Object-oriented programming

Client-Server



Client

Application that initiates peer-to-peer communication

Translate user requests into requests for data from server via protocol

GUI often used to interact with user

Server

Any program that waits for incoming communication requests from a client

Extracts requested information from data and return to client

Common Issues

- Authentication
- Authorization
- Data Security
- Privacy
- Protection
- Concurrency

What Client-Server Requires of a Programmer

Designing robust protocols

Network programming

Designing usable computer-human interfaces

Good documentation skills

Good debugging skills

Understand the information flow of the company/customer

Mastery of concurrency

Multi-platform development

Database programming

Security

Names

and abrvtns

Item	Java	Smalltalk	C#	Ruby
Class	PascalCase	PascalCase	PascalCase	PascalCase
Method	camelCase	camelCase	PascalCase	foo_bar
Field	camelCase	camelCase	CamelCase	@foo_bar
Parameter	camelCase	camelCase	camelCase	foo_bar
Local Variable	camelCase	camelCase	camelCase	foo_bar

Comments in Code

```
x = x + 1 //Add one to x
```

What does this do?

```
for i := 1 to Num do
  MeetsCriteria[ i ] := True;
for i := 1 to Num / 2 do begin
  j := i + i;
  while ( j <= Num ) do begin
    MeetsCriteria[ j ] := False;
    j := j + i;
  end;
for i := 1 to Mun do
  if MeetsCriteria[ i ] then
    writeln( i, ' meets criteria ' );
```

What does this do?

```
for PrimeCandidate:= 1 to Num do
    IsPrime[ PrimeCandidate] := True;

for Factor:= 1 to Num / 2 do begin
    FactorableNumber := Factor + Factor ;
    while ( FactorableNumber <= Num ) do begin
        IsPrime[ FactorableNumber ] := False;
        FactorableNumber := FactorableNumber + Factor ;
    end;
end;

for PrimeCandidate:= 1 to Num do
    if IsPrime[ PrimeCandidate] then
        writeln( PrimeCandidate, ' is Prime ' );
```

CVS

Concurrent Versions System

Allows multiple users to work on code

Allows access from multiple machines

References

<http://www.cvshome.org/>

<http://www.cvshome.org/docs/manual/>

<http://www.tortoisecvs.org/>

Windows client

http://www-rohan.sdsu.edu/~stremmler/CS530/AS1/remote_cvs.html

Stremmler's remote CVS on Rohan page

<http://www.eli.sdsu.edu/courses/spring05/cs580/notes/cvs/cvs.html>