COM6516
Object Oriented Programming and Software Design

The contents of this module has been developed by Adam Funk, Kirill Bogdanov, Mark Stevenson, Richard Clayton and Heidi Christensen

Practical 0

Platforms

- Java 16
- Netbeans 12.4
- jEdit

Coding style and coding standard

Comments

Platform installation

Java SE Development Kit 16

https://www.oracle.com/uk/java/technologies/javase-jdk16-downloads.html

Netbeans 12.4

https://netbeans.apache.org/download/nb124/index.html

JEdit

http://www.jedit.org/index.php

jEdit installation: JRELoadError (mac OS)

Solution: create an alias to `jedit.jar` and add this alias to the mac OS dock

Step 1: copy the icon

- Go to `/Applications` folder, right click on `jEdit.app` and select `Show Package Contents`
- Locate the file `jEdit.app/Contents/Resources/icon.icns` and open it in `Preview`
- Pick one of the icons (eg, the 3rd one down), right click on it, select `Export as…` and save it as a `jEdit.png` somewhere
- Open up the `jEdit.png` file in `Preview` and copy it (`Cmd-C`)
- Right click on the `jEdit.app/Contents/Java/jedit.jar` and select `Get Info`
- Click the icon in the top left corner of the info window — it will highlight
- Paste (`Cmd-V`) the icon onto the `jEdit.jar` file (alias will have the `jEdit` icon on it)

Step 2: create an alias

- Right click on the `jEdit.app/Contenys/Java/jedit.jar` again and select `Make Alias`
- Select the alias icon and add this as a icon on dock (`Control+Shift+Command+T`)

https://superuser.com/questions/1501724/jreloaderror-when-trying-to-open-the-jedit-text-editor-in-mac-os

Using Netbeans 1

Java quick start tutorial
https://netbeans.apache.org/kb/docs/java/quickstart.html

Java: reference guide
https://netbeans.apache.org/kb/docs/java/editor-codereference.html

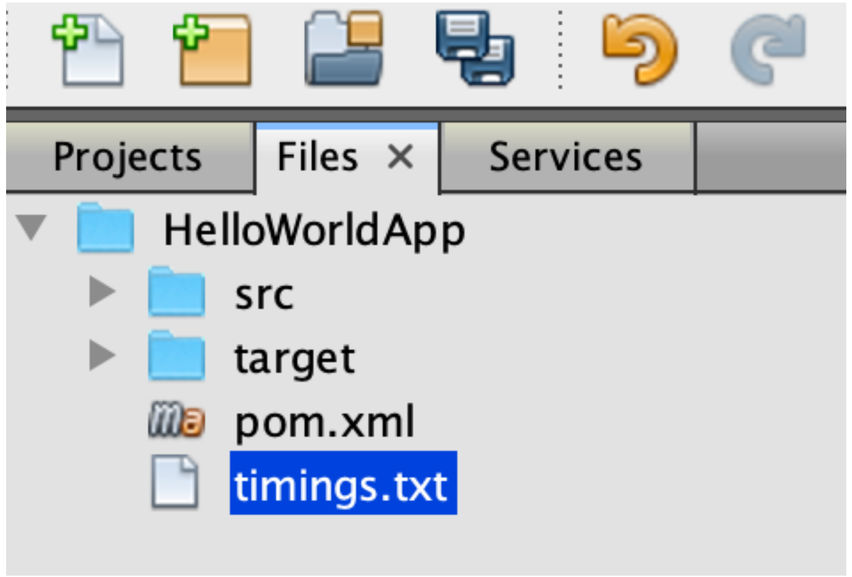
and more for Java …
https://netbeans.apache.org/kb/docs/java-se.html

A range of Netbeans tutorials
https://netbeans.apache.org/kb/docs/index.html

Using Netbeans 2

An input file is placed at the project folder

```
EasyReader inputFile = new EasyReader("timings.txt");
```



Source code starts with a '`package`' statement

```
package helloworldapp;

import sheffield.*;

/**
 * CycleComputer.java
 *
```

Using editor + command line

An input file is placed next to source code

```
EasyReader inputFile = new EasyReader("timings.txt");
```

Source code does not start with a ‘`package`’ statement

```
// package helloworldapp;

import sheffield.*;

/**
 * CycleComputer.java
 *
```

Recommended

Coding style

What is good coding style?

Coding style

```
#include <math.h>
#include <time.h>
#include <unistd.h>
#include <netinet/in.h>
typedef float F,A[3]; F D,M
[999]={ LT} ,*L=NL+M,*P,b,t,*h,*i; A
#define S(x,y,z)F x(F*d,F z s){ F t=y; t+=y; return t+=y; }
E,Q,U,V,C,c,I={ EY} ; unsigned char g[2414],*p=g,*e; int j,k,s,m
,n,x,y; S(B,*d++=*s++,*)S(o,*d+++=*s++,*)S(a,*d++-=*s++,*)S(H,*d++
**s++,*)S(X,*d++*=s,)S(v,*d+++=s,)int w(int c){ return*p++=c; } F W(F*
d){ return sqrt(H(d,d)); } void r(uint32_t u){ w(u>>24); w(u>>16); w(u>>
8); w(u); } F O(F*d){ return X(d,1/W(d)); } char*z,*f; void u(char*s){ *
s&&w((u(1+s),*s)); } F G(F x,int p){ for(z=p*46+f; 12^*z; x+=.5){ for(D=-4;
5>D; D++)if(9-*z++){ *P++=x; *P++=D/2; *P++=0; } } return p; } void K(){ write
(k,g,p-g); p=g; } F*d(){ h=0; D=1e9; a(E,Q); O(E); 2[E]&&0>(t=(1+2[Q])/2[E])&&
(D=-t,h=P); for(i=M; P>i; i+=3){ B(C,i); a(C,Q); b=H(E,C); (t=b*b-H(C,C)+(i<L?99
:.6))>=0&&0<(t=b<t?t+b:b-t)&&D>t&&(D=t,h=i); } return h; } void Y(int N){ F*h,*i
; A p,n; if(!(h=d()))*c=1[c]=(2[c]=2[E]/2)/2; else if(h<L){ X(c,0); v(c,1); } else
{ B(p,E); X(p,D); o(p,Q); if(h-P){ B(n,p); a(n,h); O(n); } else{ X(n,0); ++2[n]; }
B(Q,n); X(Q,1e-4); o(p,Q); X(c,0); if(N<8){ B(Q,n); X(Q,2*H(n,E)); a(E,Q); o(E,p);
B(Q,p); Y(1+N); X(c,h-P?.8:.2); } for(i=M; L>i; ++i){ B(E,i); B(Q,p); d()-i||v(c,(
h-P?.1:.5)*H(n,E)); } v(c,0.05); h-P||(2[c]*=.3,c[1&lrint(*p)^lrint(1[p])&1]*=.2);
} } void Z(char*s){ K(); p+=4; u(s?s:"TADI"); } void J(){ uint32_t c=~0; e=p; p=g;
r(e-p-8); while(p!=e){ c^=*p++; for(j=0; 8>j; j++)c=c/2^c%2*3988292384; } r(~c); K
(); } void q(int c){ w(c); m+=c; m%=c=65521; n+=m; n%=c; } void T(F c){ c=.5+255
*c; q(0>c?0:c>255?255:c); } struct sockaddr_in R; int main(){ time_t i; struct
tm*b; R.win_port=8224; s=socket(R.sin_family=AF_INET,SOCK_STREAM,0); bind(s,
(void*)&R, sizeof R); listen(s,1); for(; ; ){ k=accept(s,0,0); for(; ; ){ ++j;
read(k,p,1); if(*p=='\n') { if(3>j)break; j=0; } } m=1; u("\n\03Z\n\rGNP"
"\211\n\r\n\r1 :hserfeR\n\rKD 002 0.1/PTTH"); Z("RDHI"); r(800); r(600); w
(8); r(33554433); J(); Z(0); w(120); w(1); J(); i=time(0); b=localtime(&
i); x=b->tm_sec; *I=45<x?x-60:15>x?x:30-x; *U=-I[1]; 1[U]=*I; *V=2[I]*
*I; 1[V]=2[I]*1[I];2[V]=-*I**I-1[I]*1[I]; O(U); O(V); X(U,D=W(I)/
1e3); X(V,D); P=L;y=1+(11+b->tm_hour)%12; 9<y&&G(-14,y/10); G(-
10,y%10); G(-6,10); y=b->tm_min; G(-2,y/10);G(2,y%10); G(6,
10); G(10,x/10); G(14,x%10); for(z="xxxdtrb! d r y "; 9[z]; ++z){
for(y=7&8[z]; 600>y; y+=14&*z){ Z(0); w(0); p+=4; q( 0); for(x=7&9[z]; 800>x;
x+=15&1[z]){ B(Q,V); X(Q,y-300); B(E,U); X(E,x -400); o(E,Q); B(Q,I); Y(0);
T(*c); T(1[c]); T(2[c]); } j=p-g-13; 12[ g]=~(10[g]=j>>8); 11[g]=~(9[g]=
j); J(); } } Z(0); w(1); r( 65535); r(n<<16|m); n=0; J(); Z(
"DNEI"); J(); j=0; close(k); } } char*f=
```

International Obfuscated C Code Contest: http://ioccc.org/2013/mills/mills.c

Coding style

```
char rahc
[ ]
=
"\n/"
,
redivider
[ ]
=
"Able was I ere I saw elbA"
;
*
deliver,reviled
=
1+1
,
niam ; main
( )
{/*\}
\*/
int tni
=
0x0
,
rahctup,putchar
( )
,LACEDx0 = 0xDECAL,
rof ; for
(;(int) (tni);)
(int) (tni)
= reviled ; deliver =
redivider
;
for ((int)(tni)++,++reviled;reviled* *deliver;deliver++,++(int)(tni)) rof
=
(int) -1- (tni)
;reviled--;--deliver;
(tni) = (int)
- 0xDECAL + LACEDx0 -
rof ; for
(reviled--,(int)--(tni);(int) (tni);(int)--(tni),--deliver)
rahctup = putchar
(reviled* *deliver)
;
rahctup * putchar
((char) * (rahc))
;
/*\
{\*/}
```

http://ioccc.org/1987/westley.c

```
main(a,b)char**b;{int c=1,d=c,e=a-
d;for(;e;e--)_(e)<_(c)?c=e:_(e)>_(d)?
d=e:7;
while(++e<a)printf("\xe2\x96%c",129+
(**b=8*(_(e)-_(c))/(_(d)-_(c))));}
```

http://www.ioccc.org/2012/dlowe/dlowe.c

Avoid this kind of thing

http://thc.org/root/phun/unmaintain.html

Coding standards

Can be contentious — which coding standards or guidelines to use?

- Oracle coding conventions date back to 1999

https://www.oracle.com/technetwork/java/javase/overview/codeconvtoc-136057.html

- Other style guidelines (e.g.) http://geosoft.no/development/javastyle.html
- Can contribute to readability and maintainability of code

Generally accepted naming conventions

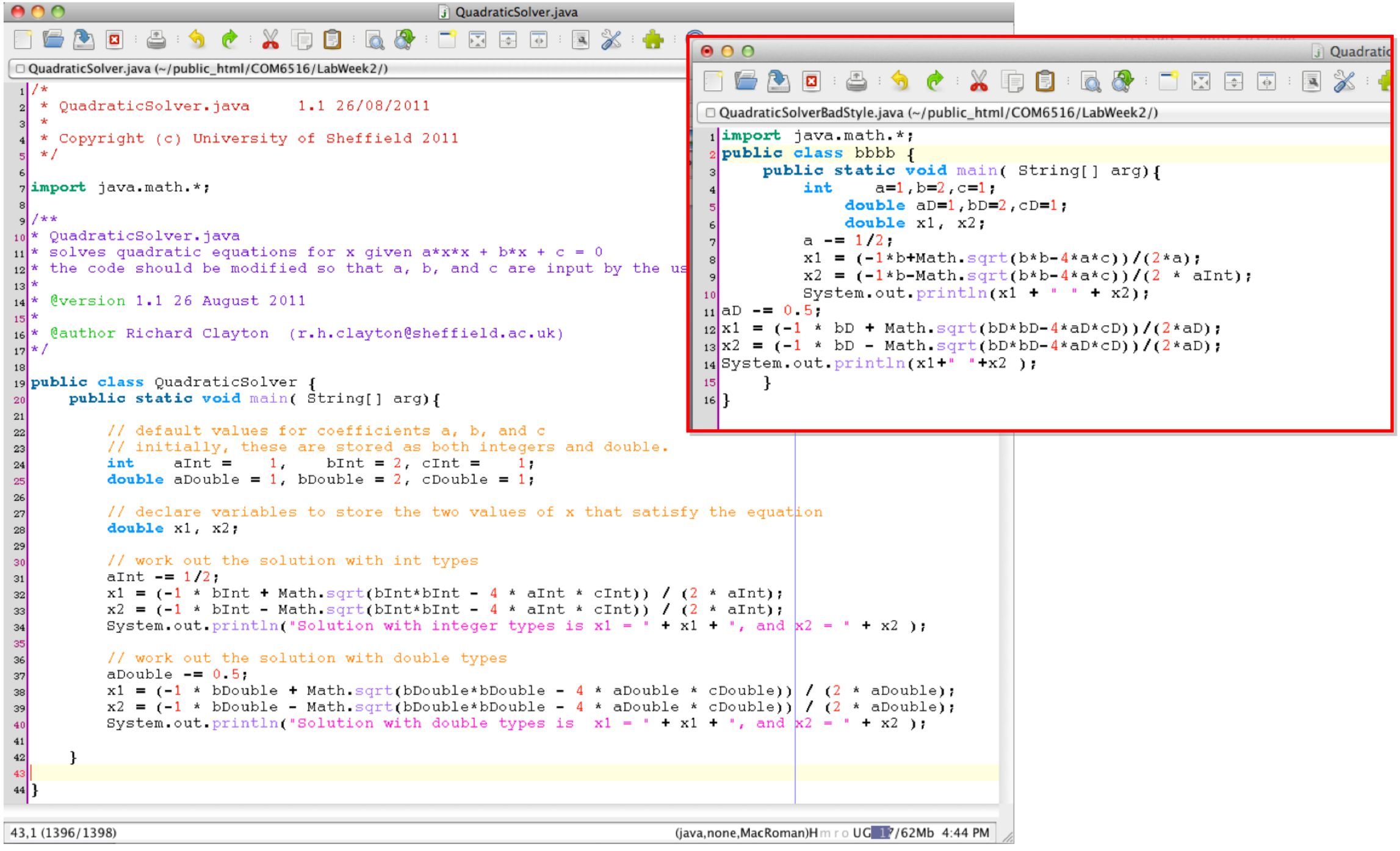
- All names in English; all comments in English 请使用英文注释
- Packages in lower case (e.g.) sheffield
- Types (classes) are nouns and Capitalised (e.g.) HelloWorld
- Variable names in mixed case, with lower case first letter (e.g.) myVariable
- Methods are verbs in mixed case. (e.g.) getMyVariable
- Constants (final variables) in upper case (e.g.) NUMBER_OF_FIELDS

Coding standards

Any violation to the guide is allowed if it enhances readability

- The main goal of the recommendation is to improve readability and thereby the understanding and the maintainability and general quality of the code. It is impossible to cover all the specific cases in a general guide and the programmer should be flexible.

https://petroware.no/html/javastyle.html


QuadraticSolver.java
QuadraticSolver.java (~/public_html/COM6516/LabWeek2/)
/*
* QuadraticSolver.java 1.1 26/08/2011
*
* Copyright (c) University of Sheffield 2011
*/
import java.math.*;
/**
* QuadraticSolver.java
* solves quadratic equations for x given a*x*x + b*x + c = 0
* the code should be modified so that a, b, and c are input by the us
*
* @version 1.1 26 August 2011
*
* @author Richard Clayton (r.h.clayton@sheffield.ac.uk)
*/
public class QuadraticSolver {
public static void main(String[] arg){
// default values for coefficients a, b, and c
// initially, these are stored as both integers and double.
int aInt = 1, bInt = 2, cInt = 1;
double aDouble = 1, bDouble = 2, cDouble = 1;
// declare variables to store the two values of x that satisfy the equation
double x1, x2;
// work out the solution with int types
aInt -= 1/2;
x1 = (-1 * bInt + Math.sqrt(bInt*bInt - 4 * aInt * cInt)) / (2 * aInt);
x2 = (-1 * bInt - Math.sqrt(bInt*bInt - 4 * aInt * cInt)) / (2 * aInt);
System.out.println("Solution with integer types is x1 = " + x1 + ", and x2 = " + x2);
// work out the solution with double types
aDouble -= 0.5;
x1 = (-1 * bDouble + Math.sqrt(bDouble*bDouble - 4 * aDouble * cDouble)) / (2 * aDouble);
x2 = (-1 * bDouble - Math.sqrt(bDouble*bDouble - 4 * aDouble * cDouble)) / (2 * aDouble);
System.out.println("Solution with double types is x1 = " + x1 + ", and x2 = " + x2);
}
}
43,1 (1396/1398)
(java,none,MacRoman)
4:44 PM
QuadraticSolverBadStyle.java (~/public_html/COM6516/LabWeek2/)
import java.math.*;
public class bbbb {
public static void main(String[] arg){
int a=1,b=2,c=1;
double aD=1,bD=2,cD=1;
double x1, x2;
a -= 1/2;
x1 = (-1*b+Math.sqrt(b*b-4*a*c))/(2*a);
x2 = (-1*b-Math.sqrt(b*b-4*a*c))/(2 * aInt);
System.out.println(x1 + " " + x2);
aD -= 0.5;
x1 = (-1 * bD + Math.sqrt(bD*bD-4*aD*cD))/(2*aD);
x2 = (-1 * bD - Math.sqrt(bD*bD-4*aD*cD))/(2*aD);
System.out.println(x1+" "+x2);
}
}

Comments

- Consistency and a minimal approach is the key to good commenting
- Choosing sensible names for classes, methods, and variables make code self-documenting
- Don’t use comments as a crutch when you can’t be bothered to structure the code carefully
- Do write your comments carefully, in good English (or whatever language is appropriate - remember for COM6516 this is English)

http://www.codinghorror.com/blog/2006/12/code-tells-you-how-comments-tell-you-why.html

https://blog.codinghorror.com/code-tells-you-how-comments-tell-you-why/

CODING HORROR

programming and human factors

Google Custom Search

18 Dec 2006

Code Tells You How, Comments Tell You Why

In an earlier post on the philosophy of code comments, I noted that **the best kind of comments are the ones you don't need**. Allow me to clarify that point. You should first strive to make your code as simple as possible to understand without relying on comments as a crutch. Only at the point where the code *cannot* be made easier to understand should you begin to add comments.

http://blog.jtimothyking.com/2006/12/15/does-bad-writing-reflect-poor-programming-skills

passes the ball to a receiver. Occasionally, the receiver is on a different page than the quarterback, and when the ball is thrown, there's no one there to catch it. The play falls apart. Players either work together, or they lose the game. In football speak, this is called "not executing well."

The same is true of programming. It was true back in the day, and it's increasingly true. Today's technologies are so complicated, no one person can know everything there is to know about any one of them. Do you "know Java"? Yeah, right. Which "Java" is that? Whichever part of Java you know, it's only a small slice of Java technology. You need to communicate with others in order to leverage their knowledge of the technologies you're using. And you need to communicate to your teammates your knowledge of the technologies you know and create. And you need to explain to others how to use your code. You need to document your own API's. Unfortunately, API documentation, both in-house and to outside programmers, stinks.

No wonder. Recruiters and hiring managers say that today's job is a knowledge-based one. And communication and teamwork skills are more important than ever. And then they go off and play buzzword bingo with a stack of resumes. (And then management commits every teamwork-killing sin in *Peopleware*, but that's a different story.) To be fair, recruiters usually know little about software development, so buzzword bingo is the only way they know how to read a resume. And software managers usually know

Currently Lead Developer at Th
Shop: Perl project consulting.

Check out more of my story.

Newest Posts

How to solve the Mac OS/X "stu
library file out of sync" proble

My Recent Posts at The Perl Sh

Subclassing vs Embedding in G

Finishing Off that Bug in django
(part trois)

Still Fixing that Bug in djangocn
(part deux)

Tags

Alistair Cockburn Apple ber
bugs communication compute