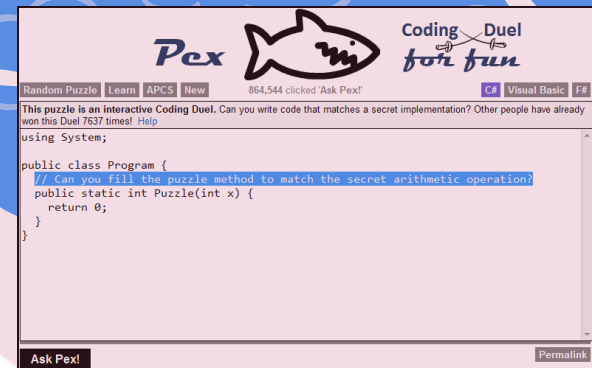




```
using System;  
  
public class Program {  
    public static int Puzzle(int x) {  
        return x * x;  
    }  
}
```

secret

<https://www.codehunt.com>



Code Hunt for University Courses

<http://pex4fun.com>

Tao Xie

University of Illinois at Urbana-Champaign

In collaboration with the Code Hunt team at Microsoft Research
and Illinois ASE students

Related Papers/Resources: <https://sites.google.com/site/asergp/projects/ese>

Educational Software Engineering

aka. Software Engineering for Education

develops **software engineering (+gaming)**
technologies for general **educational tasks**,
going beyond educational tasks for software
engineering (e.g., software engineering
education)



Software Engineering Education CFP

Theme

Interdisciplinary Topics in Software Engineering Education



Tao Xie, Nikolai Tillmann, Jonathan de Halleux, and Judith Bishop. *Educational Software Engineering: Where Software Engineering, Education, and Gaming Meet*. In Book "Computer Games and Software Engineering", Chapter 5, 2015. Earlier version: <http://web.engr.illinois.edu/~taoxie/publications/gas13-educationalse.pdf>

Yearly Code Hunt Workshop with Upcoming SE/Edu Conferences?

*International Code Hunt Workshop on
Educational Software Engineering*
as **Open** Forum for Exchanging Ideas and Early
Results of the Community



...

Your Contributions and Suggestions to Build
and Grow the Community as Follow-up to this
1st Wonderful Code Hunt Workshop!

Inspired from International SPIN Workshop on Model Checking of Software (SPIN)

Observations & Motivations of Coding Duels at Code Hunt

- Gaming/fun aspect (e.g., “reverse-engineering”)
 - Guessing/discovering/exploring, ...

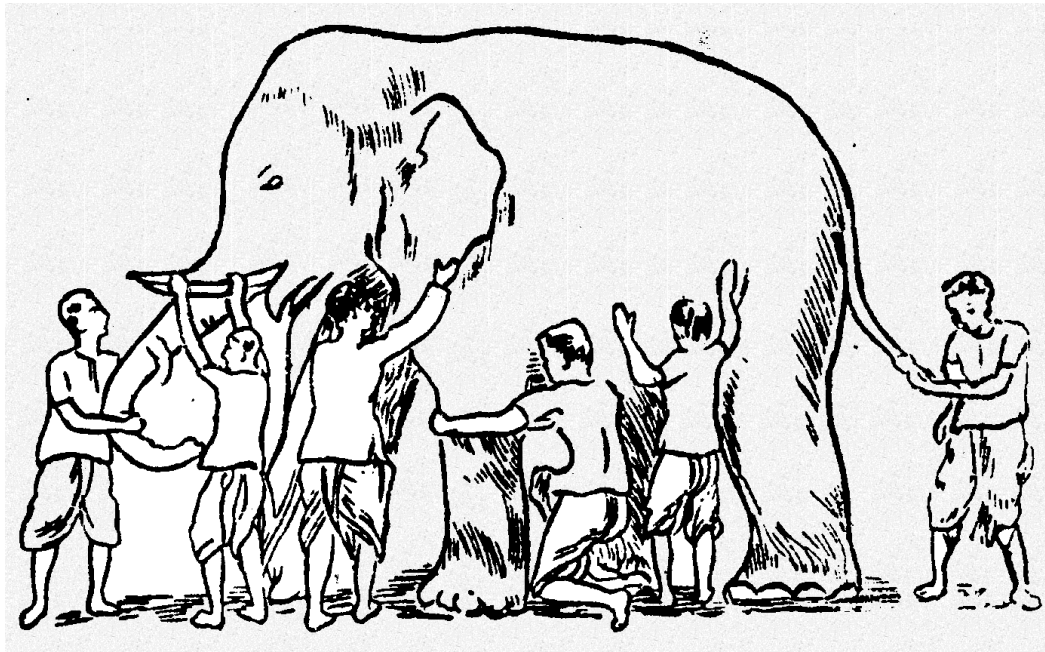


Image: <http://www.jainworld.com/literature/story25.htm>



Example User Feedback on Coding Duels

“I used to love the first person shooters and the satisfaction of blowing away a whole team of Noobies playing Rainbow Six, but this is **far more fun.**”



“I’m afraid I’ll have to **constrain** myself to spend just an hour or so a day on this **really exciting** stuff, as I’m really stuffed with work.”

“It really got me ***excited***. The part that got me most is about spreading interest in teaching CS: I do think that it’s **REALLY great for teaching | learning!**”

Observations & Motivations of Coding Duels at Code Hunt

- Gaming/fun aspect (e.g., “reverse-engineering”)
 - Guessing/discovering/exploring, ...
- Engaging/sustaining aspect
 - Not loss of courage/confidence, sense of progression
 - Earlier presented hint generation, **progress indication** with model counting

Coding Duel Competition @ICSE 2011

«ICSELevel2Challenge10» - Can you fill the puzzle method to match the secret computation? - you already made 13 attempts

Coding Duels	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29
TaoXie	3	3	2	3	3	3	5	12	6	2	20	4		13	3	2	24													
Felienne	3	7	4	4	8	10	12	17	25	10	20	7																		
RPortoAbreu	4	3	4	8	12	3	33	40	31	17	112	98	70	161	2	4	7	8	13	3	3	2	11	8	21	10	6	17	23	63
jomaras	4	2	3	4	5	9	8	10	40	8	14	6	3	17	4	2	4	9	10	3	6	2		34		8	3			
chenfucn	3	2	5	5	7	6	2	1	3		18			1																
cdragert	4	6	4	26	8	3	21	24	15	22	4	18	8	180	10	2	50	3	3	2	5	2	5	4	8	16	7	9	10	21
jamshaidm	4																													
JMacFan	2	2	2	18	12	6	18																							
Meilies	11	2	4	5	13	14	18	17		9	1																			
shauvik	2	5	2	3	5	11	6																							
almsantos	4	2	4	3	5	10	4	1		3	2	2	1	5	7	2	22	5	3	5	5	3	15	15	21	8	12		22	1
SCBSUFPE	4	5	3	4	7	7	7	21	18	3	5			14	4	2	5	6	5	3	4	3	9	15	3	11	11	15	19	7
theofour	2	2	2	4	6	12	14	4	7	14	14	37	5	58	2	2	8	6	9	8	2	2	5	4	6	7	4	7	4	12
anchi	6	4	3	4	12	15	6	15	21	7	15	28	2	7	4	2	14	5	7	4	2	3	7	3	10	6	13	17	12	62
malteres	2	5	2	3	6	5	23	18	36	10	17	36	26	224	2	4	12	5	9	2	4	8	9	2	18	15	3	17	29	79
rla4	4	2	2	4	6	7	10	22	11		9	4	12	12	7	2	11	6	8	6	2	1								
krw7c	7	6	8	3	7	7																								
MIKAND	2	2	1	5	8	5	9	7	17	5	9	7	2	9	4	2	4	4	5				5	2	5	5	5	4	13	10
ariboira	6	7	6	14	13	22	26	51	56	11	47	96	61	226	23	3	16	13	21	7	3	11	21	3	7	2	21	26	27	22
Benny	4	5	5	4	13	6	10	13	22	32	53	31	30	9	2	2	7	5	9	2	2	4	7	8	4	14	22	21	11	29
TheRama	3	2	2	3	3	18	4	8	29	7	17	10	3	37	3	2	12	7	6	2	6	2	10	5	3	11	5	22	8	12
madking	2	2	2	5	11	2	8	6	13	15	8	10	4	386	4	2	10	6	6	2	2	2	5	7	4	7	5	3	2	15
nipun	19	9	5	10	2	2	12	12		2	12			8	3	3	10	6					8	4	15				24	
(no nickname)	3	7	2	3																										
ejiadachi	6	3	2	4	7	13	17	11	9	2	3	3	24	1	3	2	3	4	12	8	3	4	6	25	8	16	6	18	134	17
schroeter	3	2	3	6	3	3	6	6	18	4	12																			
Ezzo	3	1	1																											
(no nickname)	3	4	4	11																										

#10 This is your last attempt

20 using System;

20

112 public class Program {

14 public static int Puzzle(int x) {

18 if (x <= 0) return 0;

4 if (x == 1) return 0;//1

if (x == 2) return 0;//2

if (x == 3) return 0;//3

if (x == 4) return 0;//5

if (x == 5) return 4;//7

if (x == 6) return 4;//13

if (x == 7) return 4;//21

if (x == 8) return 0;//34

if (x == 9) return 0;//55

if (x == 10) return 0;//89

if (x == 11) return 0;//144

if (x == 12) return 0;//233

if (x == 72) return (x-8);

if (x == 963) return (x-3);

if (x == 964) return (x-4);

if (x == 965) return (x-1);

if (x == 966) return (x-2);

if (x == 995) return (x-3);

if (x == 996) return (x-4);

if (x == 997) return (x-1);

if (x == 998) return (x-2);

if (x == 999) return (x-3);

return 0;

<http://pexforfun.com/icse2011>

Observations & Motivations of Coding Duels at Code Hunt

- Gaming/fun aspect (e.g., “reverse-engineering”)
 - Guessing/discovering/exploring, ...
- Engaging/sustaining aspect
 - No loss of courage/confidence, sense of progression
 - Earlier presented hint generation, **progress indication** with model counting
- Teaching/learning aspect
 - Designing coding duels for **pedagogical** purposes

Observations & Motivations of Coding Duels at Code Hunt

- Gaming/fun aspect (e.g., “reverse-engineering”)
 - Guessing/discovering/exploring, ...
- Engaging/sustaining aspect
 - No loss of courage/confidence, sense of progression
 - Earlier presented hint generation, **progress indication** with model counting
- Teaching/learning aspect
 - Designing coding duels for **pedagogical** purposes
- Classroom integration aspect
 - **Office Mix** + Code Hunt

Outline

- Progress Indication (simple techniques and some initial observations)
- Coding duels for teaching and learning
- Office Mix + Code Hunt for teaching and learning

Motivation: Why Syntactic Edit Distance Alone Not Enough

```
1 // Fancy years are years that have all the same numbers
2 // Fancy years are never less than 1000 and never bigger than 999
3 public static string isFancyYear(int i) {
4     if (i < 1000 || i > 9999)
5         return "not a fancy year";
6     int digit = i % 10;
7     while (i != 0) {
8         if (i % 10 != digit)
9             return "not a fancy year";
10        i /= 10;
11    }
12    return "fancy year";
13 }
```

```
1 public static string isFancyYear(int i) {
2     if (i == 1111) return "fancy year";
3     if (i == 2222) return "fancy year";
4     if (i == 3333) return "fancy year";
5     if (i == 4444) return "fancy year";
6     if (i == 5555) return "fancy year";
7     if (i == 6666) return "fancy year";
8     if (i == 7777) return "fancy year";
9     if (i == 8888) return "fancy year";
10    if (i == 9999) return "fancy year";
11    return "not a fancy year";
12 }
```

Example of Behavioral Subsumption

```
1 public int Reference(int x) {  
2     if (x % 7 == 0) return 8; // Conditional 1  
3     return x % 7; // Conditional 2  
4  
5  
6 }
```

Secret Program

```
1 public int Program1(int x) {  
2     if(x % 7 == 0) return 8;  
3     if(x == 1009) return 1;  
4     return x;  
5  
6 }
```

```
1 public int Program2(int x) {  
2     if(x % 7 == 0) return 8;  
3     if(x == 1009) return 1;  
4     if(x == 383) return 5;  
5     return x;  
6 }
```

Student Program 1

Subsumed by

Student Program 2

Metrics for Behavioral Similarity

Key Idea: $\# \text{Passing Paths} / (\# \text{Passing Paths} + \# \text{Failing Paths})$
Subsumption, comparison of input domain contents

- Random Sampling (RS)
- Single-program Symbolic Execution (SSE)
- Paired-program Symbolic Execution (PSE)

```
1 public void PairedProgram (object[] args) {  
2     Debug.Assert(Program1(args) == Program2(args));  
3 }
```

Advantages: simple techniques that piggy-bag Pex's existing effort/cost for supporting Code Hunt

Example: PSE works but RS and SSE do not work

```
1 int Reference(int x) {  
2  
3     return 42 - x;  
4  
5 }
```

(a) Reference Program

```
1 int Submission1(int x) {  
2  
3     return 42;  
4  
5 }
```

(b) Submission 1

```
1 int Submission2(int x) {  
2     if (x == 0) return 42;  
3     if (x == 42) return 0;  
4     return 0;  
5 }
```

(c) Submission 2

Example: SSE and PSE produce desirable metric values for behavioral similarity

```
1 public static int Puzzle(byte number) {  
2     int ct = 0;  
3     while (number != 0) {  
4         if (number % 2 > 0) ct++;  
5         number >>= 1;  
6     }  
7     return ct;  
8 }
```

(a) Reference Program

```
1 public static int Puzzle(byte number) {  
2  
3  
4     return (int) (number / 2);  
5  
6  
7  
8 }
```

(b) Submission 1

```
1 public static int Puzzle(byte number){  
2     var sum = 0;  
3     for (int i = 0; i < 8; i++)  
4     {  
5         sum += (number >> i & 1);  
6     }  
7     return sum;  
8 }
```

(c) Submission 2

Outline

- Progress Indication (simple techniques and some initial observations)
- **Coding duels for teaching and learning**
- Office Mix + Code Hunt for teaching and learning

Teaching and Learning

Pex for fun - from Microsoft Research - Page - Windows Internet Explorer

http://pexforfun.com/Page.aspx#courses/cc8e992b-efc4-4054-895b-072fb8bb6901

Search Bing

★ Favorites Pex for fun - from Microsoft Research - Page

Home RSS Email Print Page Safety Tools

The Social Classroom

Status Live Feed Edit Close



Course Description: Learn how PexForFun turns teaching computer science into a social gaming experience.

Teacher: the Pex Team

Associated Pages:

- [The Social Classroom Whitepaper](#)
- [The Social Classroom Sample Page](#)

Registered Students:

- [mbarnett](#)
- Murray
- Kai
- (no nickname)
- Chris C Sharp
- (no nickname)
- [TaoXie](#)
- (no nickname)
- meisl
- (no nickname)
- Nima
- Rocky
- Micgi
- [Ishtiaque](#)

The following link allows any signed in user to register for and access this course.
<http://pexforfun.com/thesocialclassroom>

Pex for fun



Status of The Social Classroom

Your Progress: [Nikolai Tillmann](#), you already won 1 Coding Duel; 1 more to go!

[Factorial](#) 1 attempt

 [ArraySort](#) won after 15 attempts

All Students' Progress:

Coding Duels	#0	#1
Chris C Sharp	2	2
meisl	1	1
Nima	2	
Micgi	1	

© 2010 Microsoft - Pex v0.94 - [Terms of Use](#) - [Privacy](#)

Microsoft
Research

RISE

Skills Being Trained

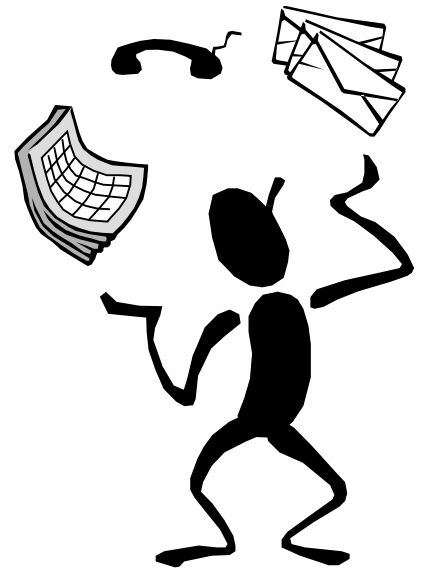
Induction

Problem solving/debugging

Program understanding/programming

Testing

Specification writing



Coding Duels for Course Assignments

@Grad Software Engineering Course



Random Puzzle Learn APCS New

1 attempt by you on this Coding Duel

C# Vi

This puzzle is an interactive Coding Duel. Can you write code that matches a secret implementation? Other people have already won this Duel 8 times! [Help](#)

```
using System;
using System.Diagnostics.Contracts;
public class Program {
    // param:pref_id:(0-201) Numeric identifier of this preference.
    // param:value:(max. 127 characters) Value of the preference to set. Set it to "0" or "" to remove this
    // preference.
    // Can you write preconditions in Code Contracts for the above natural-language requirements?
    public static int Puzzle(int pref id, string value)
    {
        return pref_id;
    }
}
```

Observed Benefits

- Automatic Grading
- Real-time Feedback (for Both Students and Teachers)
- Fun Learning Experiences

Ex Coding Duel for Training Requirements

```
public class Program {  
    // param:obj_type:non-empty Object type to add a new property to.  
    // param:prop_name:Name of the new property to add. This name needs to  
    // be a valid identifier, which is no longer than 32 characters, starting with a  
    // letter (a-z) and consisting of only small letters (a-z), numbers (0-9) and/or  
    // underscores.  
    // param:prop_type:Type of the new property: 1 for integer, 2 for string (max.  
    // 255 characters), 3 for text blob (max. 64kb)  
    // Can you write preconditions in Code Contracts for the above natural-  
    // language requirements?
```

```
public static int Puzzle(string obj_type, string prop_name, int prop_type)  
{  
    return prop_type;  
}
```

Ex Coding Duel for Training Requirements

```
public class Program {  
    // param:obj_type:non-empty Object type to add a new property to.  
    // param:prop_name:Name of the new property to add. This name needs to be a valid  
    // identifier, which is no longer than 32 characters, starting with a letter (a-z) and consisting of  
    // only small letters (a-z), numbers (0-9) and/or underscores.  
    // param:prop_type:Type of the new property: 1 for integer, 2 for string (max. 255 characters),  
    // 3 for text blob (max. 64kb)  
    public static int Puzzle(string obj_type, string prop_name, int prop_type)  
    {  
        Contract.Requires(obj_type != null );  
        Contract.Requires(obj_type.Length > 0 );  
        Contract.Requires(prop_type > 0 );  
        Contract.Requires(prop_type <= 3);  
        Contract.Requires(prop_name != null);  
        Contract.Requires(prop_name.Length > 0);  
        Contract.Requires(prop_name.Length <= 32);  
        Contract.Requires(...Regex.IsMatch(prop_name.Substring(0,1),@"[a-z]"));  
        Contract.Requires(...Regex.IsMatch(prop_name,@"^[a-z0-9]*$"));  
        return prop_type;  
    }  
}
```

Ex Coding Duel for Training Testing

```
public static string Puzzle(int[] elems, int capacity, int elem) {  
    if ((maxsize <= 0) || (elems == null) || (elems.Length > (capacity + 1)))  
        return "Assumption Violation!";  
    Stack s= new Stack(capacity);  
    for (int i = 0; i < elems.Length; i++)  
        s.Push(elems[i]);  
    int origSize = s.GetNumOfElements();  
    //Please fill in below test scenario on the s stack  
  
    //The lines below include assertions to assert the program behavior  
    PexAssert.IsTrue(s.GetNumOfElements() == origSize + 1);  
    PexAssert.IsTrue(s.Top() == elem); PexAssert.IsTrue(!s.IsEmpty());  
    PexAssert.IsTrue(s.IsMember(elem));  
    return s.GetNumOfElements().ToString() + "; " + s.Top().ToString() + "; "  
        + s.IsMember(elem).ToString() + "; " + s.IsEmpty();  
}
```

Set up a stack with some elements

Cache values used in assertions

Ex Coding Duel for Training Testing

```
public static string Puzzle(int[] elems, int capacity, int elem) {  
    if ((maxsize <= 0) || (elems == null) || (elems.Length > (capacity + 1)))  
        return "Assumption Violation!";  
    Stack s= new Stack(capacity);  
    for (int i = 0; i < elems.Length; i++)  
        s.Push(elems[i]);  
    int origSize = s.GetNumOfElements();  
    //Please fill in below test scenario on the s stack  
  
    if (s.IsMember(elem)) return "Assumption Violation!";  
    if (s.GetNumberOfElements() >= s.MaxSize()) return "Assumption Violation!";  
    s.Push(elem);  
  
    //The lines below include assertions to assert the program behavior  
    PexAssert.IsTrue(s.GetNumOfElements() == origSize + 1);  
    PexAssert.IsTrue(s.Top() == elem); PexAssert.IsTrue(!s.IsEmpty());  
    PexAssert.IsTrue(s.IsMember(elem));  
    return s.GetNumOfElements().ToString() + "; " + s.Top().ToString() + "; "  
        + s.IsMember(elem).ToString() + "; " + s.IsEmpty(); }  
}
```


Coding Duel 1 for Training Design Patterns

Can you complete method bodies of some specified methods of the given classes to match the secret implementation?

```
public class DesignPatternEx {  
    //Please fill in the method bodies of CreatePages() declared on Document1 and  
    Document2  
    //to match the behavior of the secret implementation.  
    //Hint: the classes below realize a design pattern  
    public static string Puzzle(int[] wordNumsPerPage) {  
        Document[] documents = new Document[2];  
        documents[0] = new Document1();  
        documents[1] = new Document2();  
        if (wordNumsPerPage == null) return "Assumption Violation!";  
        if (wordNumsPerPage.Length != (documents[0].Pages.Count +  
                                         documents[1].Pages.Count)) return "Assumption Violation!";  
        int totalFee = 0; int pageIndex = 0;  
        foreach (Document document in documents) {  
            foreach (Page page in document.Pages) {  
                totalFee = totalFee + page.CalPageFee(wordNumsPerPage[pageIndex++]); } }  
        return totalFee.ToString(); } }
```

Coding Duel 1 for Training Design Patterns

```
class Document1 : Document {  
    public override void CreatePages() {  
        //fill in code here  
    }  
}
```

```
class Document2 : Document {  
    public override void CreatePages() {  
        //fill in code here  
    }  
}
```

//abstract classes

```
abstract class Page {  
    abstract public int CalPageFee(int wordCount); }
```

```
abstract class Document {  
    private List<Page> _pages = new List<Page>();  
    public Document() { this.CreatePages(); }  
    public List<Page> Pages { get { return _pages; } }  
    public abstract void CreatePages(); }
```

```
class PageType1 : Page {  
    public override int CalPageFee(int  
wordCount) { return wordCount * 1; } }
```

```
class PageType2 : Page { public override int  
CalPageFee(int wordCount) { return  
wordCount * 2; } }
```

```
class PageType3 : Page { public override int  
CalPageFee(int wordCount) { return  
wordCount * 3; } }
```

```
class PageType4 : Page { public override int  
CalPageFee(int wordCount) { return  
wordCount * 4; } }
```

```
class PageType5 : Page { public override int  
CalPageFee(int wordCount) { return  
wordCount * 5; } }
```

Coding Duel 1 for Training Design Patterns

```
class Document1 : Document {  
    public override void CreatePages() {  
        Pages.Add(new PageType1());  
        Pages.Add(new PageType3());  
        Pages.Add(new PageType4());  
    }  
}
```

```
class Document2 : Document {  
    public override void CreatePages() {  
        Pages.Add(new PageType2());  
        Pages.Add(new PageType5());    }  
}
```

```
class PageType1 : Page {  
    public override int CalPageFee(int wordCount) { return  
        wordCount * 1; } }
```

```
class PageType2 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 2; } }
```

```
class PageType3 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 3; } }
```

```
class PageType4 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 4; } }
```

```
class PageType5 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount* 5; } }
```

//abstract classes

```
abstract class Page {  
    abstract public int CalPageFee(int wordCount); }
```

```
abstract class Document {  
    private List<Page> _pages = new List<Page>();  
    public Document() { this.CreatePages(); }  
    public List<Page> Pages { get { return _pages; } }  
    public abstract void CreatePages(); }
```

Coding Duel 1 for Training Design Patterns

```
class Document1 : Document {  
    public override void CreatePages() {  
        Pages.Add(new PageType1());  
        Pages.Add(new PageType3());  
        Pages.Add(new PageType4());  
    }  
}
```

```
class Document2 : Document {  
    public override void CreatePages() {  
        Pages.Add(new PageType2());  
        Pages.Add(new PageType5());    }  
}
```

```
class PageType1 : Page {  
    public override int CalPageFee(int wordCount) { return  
        wordCount * 1; } }
```

```
class PageType2 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 2; } }
```

```
class PageType3 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 3; } }
```

```
class PageType4 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount * 4; } }
```

```
class PageType5 : Page { public override int  
    CalPageFee(int wordCount) { return wordCount* 5; } }
```

//abstract classes

```
abstract class Page {  
    abstract public int CalPageFee(int wordCount); }
```

```
abstract class Document {  
    private List<Page> _pages = new List<Page>();  
    public Document() { this.CreatePages(); }  
    public List<Page> Pages { get { return _pages; } }  
    public abstract void CreatePages(); }
```

Coding Duel 3 for Training Design Patterns

```
public class DesignPatternEx
{
    //Please fill in the method bodies of GetTotalFee() and CalPageFee(int) declared
on CalPageFeeSystem to match the behavior of the secret implementation.
    //Hint: the classes below realize a design pattern
    public static string Puzzle(int[] wordNumsPerPage)
    {
        if (wordNumsPerPage == null)
            return "Assumption Violation!";
        if (wordNumsPerPage.Length != 5)
            return "Assumption Violation!";
        CalPageFeeSystem calPageFeeSystem = new CalPageFeeSystem();
        calPageFeeSystem.CalPageFee(wordNumsPerPage[0]);
        calPageFeeSystem.CalPageFee(wordNumsPerPage[1]);
        calPageFeeSystem.CalPageFee(wordNumsPerPage[2]);
        calPageFeeSystem.CalPageFee(wordNumsPerPage[3]);
        calPageFeeSystem.CalPageFee(wordNumsPerPage[4]);
        return calPageFeeSystem.GetTotalFee().ToString();    } }
```

Coding Duel 3 for Training Design Patterns

```
class CalPageFeeSystem
{
    private CalPageFeeByType _calPageFeeByType;
    public CalPageFeeSystem()
    {
        _calPageFeeByType = new CalPageFeeByType1(0, this);
    }

    public CalPageFeeByType CalPageFeeByType
    {
        get { return _calPageFeeByType; }
        set { _calPageFeeByType = value; }
    }

    public int GetTotalFee()
    {
        //fill in code here by replacing the line below

    }

    public void CalPageFee(int wordCount)
    {
        //fill in code here

    }
}
```

Coding Duel 3 for Training Design Patterns

```
class CalPageFeeByType1 : CalPageFeeByType
{
    public CalPageFeeByType1(int totalFee, CalPageFeeSystem sys)
    { this.TotalFee = totalFee;  this.CalSystem = sys;  }

    public override void CalPageFee(int wordCount)
    {
        this.TotalFee =  this.TotalFee + wordCount * 1;
        this.CalSystem.CalPageFeeByType = new CalPageFeeByType2(this.TotalFee,
this.CalSystem);  }  }
```

```
class CalPageFeeByType2 : CalPageFeeByType
{
    public CalPageFeeByType2(int totalFee, CalPageFeeSystem sys)
    { this.TotalFee = totalFee;  this.CalSystem = sys;  }

    public override void CalPageFee(int wordCount)
    { this.TotalFee =  this.TotalFee + wordCount * 2;
        this.CalSystem.CalPageFeeByType = new CalPageFeeByType3(this.TotalFee,
this.CalSystem);  }  }
```

Coding Duel 3 for Training Design Patterns

```
class CalPageFeeByType3 : CalPageFeeByType
{
    public CalPageFeeByType3(int totalFee, CalPageFeeSystem sys)
    {
        this.TotalFee = totalFee;    this.CalSystem = sys;    }

    public override void CalPageFee(int wordCount)
    {
        this.TotalFee =  this.TotalFee + wordCount * 3;
        this.CalSystem.CalPageFeeByType = new CalPageFeeByType4(this.TotalFee,
this.CalSystem);    }    }
```

```
class CalPageFeeByType4 : CalPageFeeByType
{
    public CalPageFeeByType4(int totalFee, CalPageFeeSystem sys)
    {
        this.TotalFee = totalFee;    this.CalSystem = sys;    }

    public override void CalPageFee(int wordCount)
    {
        this.TotalFee =  this.TotalFee + wordCount * 4;
        this.CalSystem.CalPageFeeByType = new CalPageFeeByType5(this.TotalFee,
this.CalSystem);
    }    }
```


Coding Duel 3 for Training Design Patterns

```
class CalPageFeeSystem
{
    private CalPageFeeByType _calPageFeeByType;
    public CalPageFeeSystem()
    {
        _calPageFeeByType = new CalPageFeeByType1(0, this);
    }

    public CalPageFeeByType CalPageFeeByType
    { get { return _calPageFeeByType; } set { _calPageFeeByType = value; } }

    public int GetTotalFee()
    {
        //fill in code here by replacing the line below
        return _calPageFeeByType.GetTotalFee();
    }

    public void CalPageFee(int wordCount)
    {
        //fill in code here
        _calPageFeeByType.CalPageFee(wordCount);
    }
}
```

Reminder: Coding Duels ARE Powerful/Flexible

<http://parsonspuzzles.appspot.com/>

ParsonsPuzzles.com

Latest Puzzle Collections

Java Algorithms

These don't use Turtles or Pictures. They are meant to complement instruction that is more visual, by getting students accustomed to standard word problems.

Algorithmic Puzzles

Test your knowledge on binary tree traversal and sorting algorithms.

Latest Puzzles

Make (Canadian) Change

Arrange pseudocode for a Canadian 'makeChange' method that takes two parameters: the amount owed and the amount paid. The method should print out how to give change (the number of dollars, the number of quarters, the number of dimes, and the number of nickels). Canada has recently phased out use of the penny, so in making change if the amount owed is \$3.72, it is rounded down to \$3.70. If the amount owed is \$3.73, it is rounded up to \$3.75. You need to make sure that the amount owed is positive, that the amount paid is positive, and that the amount paid is equal to or greater than the amount owed. If any of those things is not true, an error message should be printed and the method should return.

test2a

Make a turtle method that draws a line between two sets of coordinates, in the specified color and with the specified line thickness.

parameter example

Draw a line from the first set of coordinates to the second set of coordinates

Demo 2

ddd

Find Max

In these puzzles, you should drag and drop lines to construct a function that does what the name suggests. Lines should be sorted into correct order.

...

Amenable to Touch-based Devices for
Moving Objects/Lines ... Hololens?



Outline

- Progress Indication (simple techniques and some initial observations)
- Coding duels for teaching and learning
- **Office Mix + Code Hunt for teaching and learning**

Office Mix + Code Hunt

<https://mix.office.com/>

<https://mix.office.com/watch/4ibulo2jy5tp>

**Office Mix**
Customer Preview

Gallery My Mixes Tao Xie ▾



<https://www.codehunt.com>

**Teaching and Learning
Programming and
Software Engineering
via Interactive Gaming**



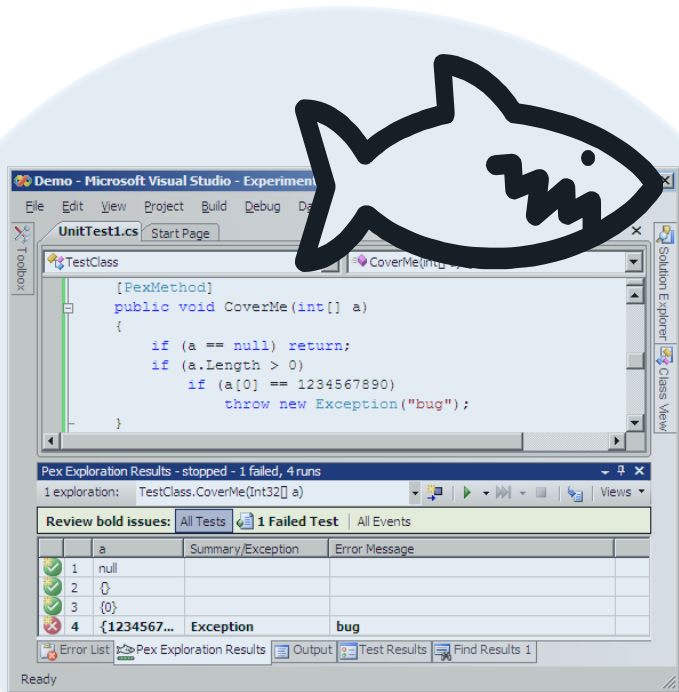
<http://pex4fun.com>



Tao Xie
University of Illinois at Urbana-Champaign



Summary: Testing Tool → Educational Gaming



DSE/Pex

Support



Code Hunt/Pex for Fun:

Interactive Gaming for

Teaching and Learning

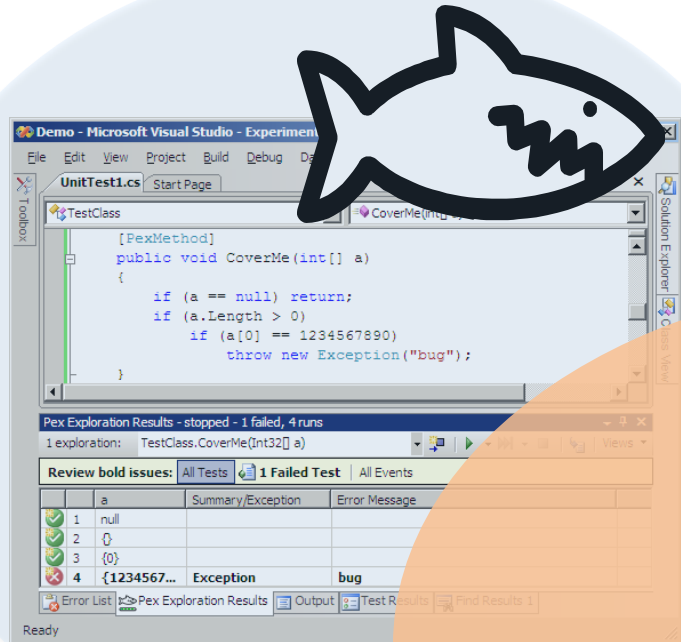
<https://www.codehunt.com>

<http://pex4fun.com>

Testing Tool → Educational Gaming

<http://research.microsoft.com/pex/>

<http://pex4fun.com>



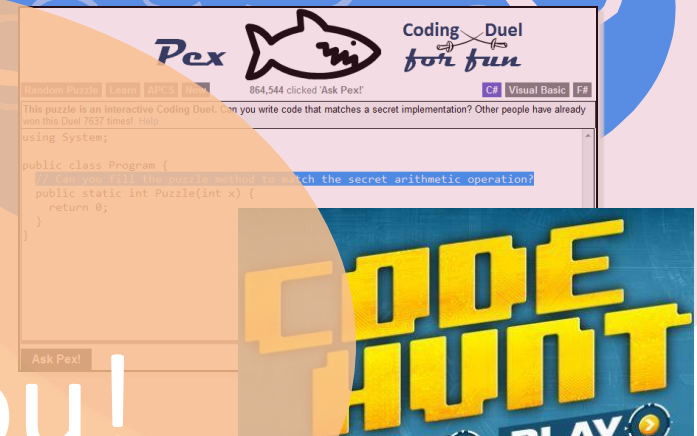
Support



```
using System;

public class Program {
    public static int Puzzle(int x) {
        return x * x;
    }
}
```

secret



Q & A

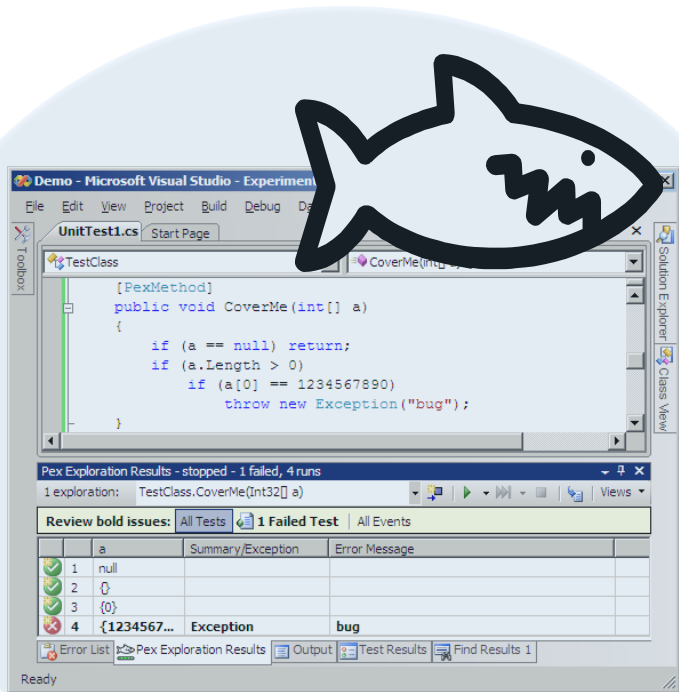
Thank you!

Contact: taoxie@illinois.edu

<http://www.cs.illinois.edu/homes/taoxie/>

Related Papers/Resources: <https://sites.google.com/site/asergpr/projects/ese>

Summary: Testing Tool → Educational Gaming



DSE/Pex

Support



Code Hunt/Pex for Fun:

Interactive Gaming for

Teaching and Learning

<https://www.codehunt.com>

<http://pex4fun.com>