Testing Your Interpreter, Part 5

Here are the tests for your interpreter, part 5.

Test 1 should return 20 when running A's main.

```
class A {
  static var x = 10;
  static function main() {
    return A.x + x;
  }
}
```

Test 2 should return 400 when running Square's main.

```
class Rectangle {
  static var width = 10;
  static var height = 12;

  static function area() {
    var a = width * height;
    return a;
  }

  static function setSize(x, y) {
    width = x;
    height = y;
  }
}

class Square extends Rectangle {
  static function setSize(x) {
    super.setSize(x, x);
  }

  static function main() {
    setSize(20);
    return area();
  }
}
```

Test 3 should return 530 when running B's main.

```
class A {
  static var x = 10;
  static var y = 20;

  static function add(a, b) {
    return a + b;
  }

  static function main() {
    return A.add(x, A.y);
  }
}

class B extends A {
  static var y = 200;
  static var z = 300;

  static function main() {
    return add(B.x+A.y,B.z+y);
  }
}
```

Test 4 should return 615 when running B's main.

```
class A {
  static var a = 1;
  static var b = 10;

  static function getSum() {
    return a + b;
  }
}

class B {
  static function main() {
    A.a = 5;

    return A.getSum() + C.x + C.timesX(A.a);
  }
}

class C {
  static var x = 100;
  static function timesX(a) {
    return a * x;
  }
}
```

Test 5 should return -716 when run with C's main.

```
class A {
  static var count = 0;

  static function subtract(a, b) {
    if (a < b) {
       throw b - a;
    }
    else
       return a - b;
  }
}

class B extends A {
  static function divide(a, b) {
    if (b == 0)
      throw a;
    else
      return a / b;
  }

  static function reduce(a, b) {
    while (a > 1 || a < -1) {
      try {
        a = divide(a, b);
        if (a == 2)
          break;
      }
      catch (e) {
        return subtract(a, b);
      }
      finally {
        count = count + 1;
      }
    }
    return a;
  }
}

class C {
  static function main() {
    var x;

    try {
      x = B.reduce(10, 5);
      x = x + B.reduce(81, 3);
      x = x + B.reduce(5, 0);
      x = x + B.reduce(-2, 0);
      x = x + B.reduce(12, 4);
    }
    catch (a) {
      x = x * a;
    }
    finally {
      x = -1 * x;
    }
    return x - A.count * 100;
  }
}
```

Test 6 should return 15 when running A's main.

```
class A {
  var x = 5;
  var y = 10;

  static function main() {
    var a = new A();
    return a.x + a.y;
  }
}
```

Test 7 should return 12 when running A's main.

```
class A {
  function add(g, h) {
    return g + h;
  }

  static function main() {
    var a = new A();
    return a.add(10, 2);
  }
}
```

Test 8 should return 110 when running A's main.

```
class A {

  var x = 100;
  var y = 10;

  function add(g, h) {
    return g + h;
  }

  static function main() {
    var a = new A();
    return a.add(a.x, a.y);
  }
}
```

Test 9 should return 125 when running A's main.

```
class A {

  var x = 100;

  function add(x) {
    return this.x + x;
  }

  static function main() {
    var a = new A();
    return a.add(25);
  }
}
```

Test 10 should return 36 when running A's main.

```
class A {

  var x = 100;

  function setX(x) {
    this.x = x;
  }

  function add(a) {
    return a.x + this.x;
  }

  static function main() {
    var a1 = new A();
    var a2 = new A();
    a1.setX(30);
    a2.setX(6);
    return a1.add(a2);
  }
}
```

Test 11 should return 54 when running A's main.

```
class A {

  var x = 100;

  function setX(x) {
    this.x = x;
  }

  function getX() {
    return this.x;
  }

  function add(a) {
    return a.getX() + this.getX();
  }

  static function main() {
    var a1 = new A();
    var a2 = new A();
    a1.setX(50);
    a2.setX(4);
    return a1.add(a2);
  }
}
```

Test 12 should return 26 when running C's main.

```
class A {
  var x = 1;
  var y = 2;

  function m() {
    return this.m2();
  }

  function m2() {
    return x+y;
  }
}

class B extends A {
  var y = 22;
  var z = 3;

  function m() {
    return super.m();
  }

  function m2() {
    return x+y+z;
  }
}

class C extends B {
  var y = 222;
  var w = 4;

  function m() {
    return super.m();
  }

  static function main() {
    var c = new C();
    return c.m();
  }
}
```

Test 13 should return 117 when running Square's main.

```
class Shape {
  function area() {
    return 0;
  }
}

class Rectangle extends Shape {
  var height;
  var width;

  function setHeight(h) {
    height = h;
  }

  function setWidth(w) {
    width = w;
  }

  function getHeight() {
    return height;
  }

  function getWidth() {
    return width;
  }

  function area() {
    return getWidth() * getHeight();
  }
}
```

```
class Square extends Rectangle {
  function setSize(size) {
    super.setWidth(size);
  }

  function getHeight() {
    return super.getWidth();
  }

  function setHeight(h) {
    super.setWidth(h);
  }

  static function main() {
    var s = new Square();
    var sum = 0;
    s.setSize(10);
    sum = sum + s.area();
    s.setHeight(4);
    sum = sum + s.area();
    s.setWidth(1);
    sum = sum + s.area();
    return sum;
  }
}
```

Test 14 should return 32 when running Square's main.

```
class Shape {
  function area() {
    return 0;
  }

  function largerThan(s) {
    return this.area() > s.area();
  }
}

class Rectangle extends Shape {
  var height;
  var width;

  function setHeight(h) {
    height = h;
  }

  function setWidth(w) {
    width = w;
  }

  function getHeight() {
    return height;
  }

  function getWidth() {
    return width;
  }
  function area() {
    return getWidth() * getHeight();
  }
}

class Square extends Rectangle {
  function setSize(size) {
    super.setWidth(size);
  }

  function getHeight() {
    return super.getWidth();
  }

  function setHeight(h) {
    super.setWidth(h);
  }

  static function main() {
    var s1 = new Square();
    var s2 = new Rectangle();
    var s3 = new Square();
    s1.setSize(5);
    s2.setHeight(8);
    s2.setWidth(4);
    s3.setWidth(3);

    var max = s1;
    if (s2.largerThan(max))
      max = s2;
    if (s3.largerThan(max))
      max = s3;

    return max.area();
  }
}
```

Test 15 should return 15 when running List's main.

```
class List {
  var val;
  var next;

  function getNext() {
    return next;
  }

  function setNext(x) {
    if (x == 0)
      next = 0;
    else {
      next = new List();
      next.setVal(val+1);
      next.setNext(x-1);
    }
  }

  function setVal(x) {
    val = x;
  }

  static function main() {
    var l = new List();
    l.setVal(10);
    l.setNext(5);
    return l.getNext().getNext().getNext().getNext().getNext().val;
  }
}
```

Test 16 should return 16 when running Box's main.

```
class Box {
  static var countAccesses = 0;
  var size = 1;

  function setSize(s) {
    this.size = s;
    countAccesses = countAccesses + 1;
  }

  static function main() {
    var x = 0;
    var c;

    while (x < 10) {
      var a = new Box();
      a.setSize(x + countAccesses);
      if (a.size % 4 == 0)
        c = a;
      x = x + 1;
    }

    return c.size;
  }
}
```

Test 17 should return 123456 when running List's main

```
class List {
  var val;
  var next;

  function getNext() {
    return next;
  }

  function setNext(next) {
    this.next = next;
  }

  function makeList(x) {
    if (x == 0)
      next = 0;
    else {
      next = new List();
      next.setVal(val+1);
      next.makeList(x-1);
    }
  }

  function setVal(x) {
    val = x;
  }

  function reverse() {
    if (getNext() == 0)
      return this;
    else
      return getNext().reverse().append(this);
  }

  function append(x) {
    var p = this;
    while (p.getNext() != 0)
      p = p.getNext();
    p.setNext(x);
    x.setNext(0);
    return this;
  }

  static function main() {
    var l = new List();
    l.setVal(1);
    l.makeList(5);
    l = l.reverse();

    var result = 0;
    var p = l;
    var c = 1;
    while (p != 0) {
      result = result + c * p.val;
      c = c * 10;
      p = p.getNext();
    }
    return result;
  }
}
```

Test 18 should return 5285 when running List's main.

```
class List {
  var val;
  var next;

  function getNext() {
    return next;
  }

  function makeList(x) {
    if (x == 0)
      next = 0;
    else {
      next = new List();
      next.setVal(getVal()+1);
      next.makeList(x-1);
    }
  }
  function setVal(x) {
    val = x;
  }

  function getVal() {
    return val;
  }

  function expand() {
    var p = this;
    while (p != 0) {
      function exp(a) {
        while (a != 0) {
          this.setVal(this.getVal() + p.getVal() * a.getVal());
          a = a.getNext();
        }
      }
      exp(p);
      p = p.getNext();
    }
  }


  static function main() {
    var l = new List();
    l.val = 1;
    l.makeList(5);
    l.expand();
    return l.getVal();
  }
}
```

Test 19 should return 100 when running A's main.

```
class A {
  static function divide(x, y) {
    if (y == 0)
      throw new Zero();
    return x / y;
  }

  static function main() {
    var x;

    try {
      x = divide(10, 5) * 10;
      x = x + divide(5, 0);
    }
    catch(e) {
      x = e.getValue();
    }
    finally {
      x = x + 100;
    }
    return x;
  }
}

class Zero {
  var value = 0;

  function getValue() {
    return value;
  }
}
```

Test 20 should return 420 when running A's main.

```
class A {
  static function divide(x, y) {
    if (y == 0)
      throw new Zero();
    return x / y;
  }

  static function main() {
    var x = 0;
    var j = 1;

    try {
     while (j >= 0) {
      var i = 10;
      while (i >= 0) {
        try {
          x = x + divide(10*i, i);
        }
        catch(e) {
          x = x + divide(e.getValue(), j);
        }
        i = i - 1;
      }
      j = j - 1;
     }
    }
    catch (e2) {
      x = x * 2;
    }
    return x;
  }
}

class Zero {
  var value = 10;

  function getValue() {
    return value;
  }
}
```

Test 21 should return 10 when running A's main.

```
class A {
  var value = 10;

  static function main() {
    try {
      var a = new A();
      a.value = 100;
      throw new A();
      return a.value;
    }
    catch (a) {
      return a.value;
    }
  }
}
```

Test 22 should throw an error for "no this".

```
class A {
  var x = 10;

  static function nowork(x) {
    return this.x;
  }

  function mightwork() {
    return x + nowork(x);
  }

  static function main() {
    var a = new A();
    return a.mightwork();
  }
}
```

Tests for those who did past challenges: overloaded functions, call-by-reference, expressions with side-effects

Test 31 should return 530 when running A's main.

```
class A {
  static var x = 10;
  static var y = 20;

  static function add(a, b) {
    return a + b;
  }

  static function add(a,b,c) {
    return a + b + c;
  }

  static function main() {
    return A.add(x, y) + A.add(x, y, y) * 10;
  }
}
```

Test 32 should return 66 when running B's main.

```
class A {
  static var x = 10;
  static var y = 20;

  static function add(a, b) {
    return a + b;
  }

  static function add(a,b,c) {
    return a + b + c;
  }
}

class B extends A {
  static var x = 2;
  static var y = 30;

  static function add(a,b) {
    return a*b;
  }

  static function main() {
    return add(x,y) + add(x,x,x);
  }
}
```

Test 33 should return 1026 when running A's main.

```
class A {
  static var x = 5;

  static function swap(& a, & b) {
    var temp = a;
    a = b;
    b = temp;
  }

  static function main() {
    var y = 10;
    var sum = 0;
    swap(x, y);
    sum = x * 100 + y;
    x = 1;
    y = 2;
    swap(A.x, y);
    sum = sum + x * 10 + y;
    return sum;
  }
}
```

Test 34 should return 2045 when running A's main.

```
class A {
  static var x = 0;

  static function setSum(limit) {
    var sum = 0;
    while ((x = x + 1) < limit) {
      sum = sum + x;
    }
    return sum;
  }

  static function main () {
    var j = setSum(10);
    return (x * 200 + j);
  }
}
```

Tests for those seeking the challenge of non-default constructors

Test 41 should return 417 when running Square's main.

```
class Shape {
  function area() {
    return 0;
  }
}

class Rectangle extends Shape {
  var height;
  var width;

  Rectangle(h, w) {
    this.height = h;
    this.width = w;
  }

  function setHeight(h) {
    height = h;
  }

  function setWidth(w) {
    width = w;
  }

  function getHeight() {
    return height;
  }

  function getWidth() {
    return width;
  }

  function area() {
    return getWidth() * getHeight();
  }
}

class Square extends Rectangle {
  Square(size) {
    super(size, size);
  }

  function setSize(size) {
    super.setWidth(size);
  }
  function getHeight() {
    return super.getWidth();
  }

  function setHeight(h) {
    super.setWidth(h);
  }

  static function main() {
    var s = new Square(20);
    var sum = 0;
    sum = sum + s.area();
    s.setHeight(4);
    sum = sum + s.area();
    s.setWidth(1);
    sum = sum + s.area();
    return sum;
  }
}
```

Test 42 should return 10 when running A's main.

```
class A {
  var x;

  A(val) {
    x = val;
  }

  static function main() {
    var a = new A(10);
    return a.x;
  }
}
```

Test 43 should return 48 when running B's main.

```
class A {
  var x;
  var y;

  A() {
    x = 10;
    y = 2;
  }
}

class B extends A {
  var factor;

  B(f) {
    factor = f;
  }

  static function main() {
    var b = new B(4);
    return b.factor * (b.x + b.y);
  }
}
```

Test 44 should return 1629 when running A's main.

```
class A {
  var x = 1;
  var y = x + 1;
  var z = x + y + 1;

  A(a) {
    x = a;
  }

  A(a, b) {
    x = a;
    y = b;
  }

  static function main() {
    var a = new A(10);
    var b = new A(20, 5);

    return (a.x + a.y + a.z) * 100 + (b.x + b.y + b.z);
  }
}
```