NAME______________________

Exam 1
CSCI 2600 Principles of Software
September 26, 2014

- DO NOT OPEN THIS EXAM UNTIL TOLD TO DO SO!
- READ THROUGH THE ENTIRE EXAM BEFORE STARTING TO WORK.
- YOU ARE ALLOWED ONLY 1 “CHEAT” PAGES. NO OTHER MATERIAL IS ALLOWED.

This exam is worth 100 points.

Make sure you have 8 pages counting this one. There are 3 parts, each including multiple questions. If you need more room for an answer than is provided, please use the back of the page and indicate that you have done so. If you re-do a question, please make clear what is your final answer.

Be clear and brief in your explanations—rambling and lengthy answers will be penalized. All questions have short answers.

The following is for the use of graders

1. ________/50
2. ________/30
3. ________/20

TOTAL:__________/100
Part I. Reasoning About Code

Note: All variables are integers

1) Order the conditions from strongest to weakest

(a) $x$ is odd $\&$ $y = x + 1$
(b) $y$ is not odd
(c) $x$ is odd $\&$ $y$ is even
(d) $x = 11$ $\&$ $y = 10$

Answer: __, __, __, __

(a) $y \geq 0$ $\&$ $y \leq 10$
(b) $y \geq -10$ $\&$ $y \leq 10$
(c) $y = 5$
(d) $y$ is an odd $\&$ $y \geq 1$ $\&$ $y \leq 5$

Answer: __, __, __, __

2) Are the following Hoare triples true or false? Circle one.

a) TRUE / FALSE $\{ x > 0$ $\&$ $y > 0 \}$ $z = x + y$ $\{ z \geq 0 \}$

b) TRUE / FALSE $\{ x > 0$ $\&$ $y < 0 \}$ $z = x + y$ $\{ z = 0 \}$

c) TRUE / FALSE $\{ x = 5$ $\&$ $y < -5 \}$ $z = x + y$ $\{ z \leq 0 \}$

d) TRUE / FALSE $\{ x \neq y \}$ if $\{ y < x \}$ $\{ z = x$; $x = y$; $y = z$; $\}$ $\{ x > y \}$

3) Compute the weakest precondition. Fill in all intermediate conditions.

a) $\{ \}$
   if $\{ x > y \}$ $\{
   \{
   x++;$
   \{
   \}
   \} \}
else $\{
\{
 y = x + y;$
\{
\}
\}$
$\{ x > y \}$

CSCI 2600 Exam 1, Fall 2014
b) \[
\begin{array}{l}
\{ \ \ \ } \\
if ( \ x == 0 \ ) \ { \\
\{ \ \ \ } \\
res = 0; \\
\{ \ \ \ } \\
}\}
\]
else \ {
\{ \ \ \ } \\
if ( x > 0 ) \ { \\
\{ \ \ \ } \\
x = -x; \\
\{ \ \ \ } \\
}\}
else \ {
\{ \ \ \ } \\
// Do nothing; \\
\{ \ \ \ } \\
\}
\{ \ \ \ } \\
res = x*x*x; \\
\{ \ \ \ } \\
\}
\{res < -1 \}
3) Prove that `trailingZeros(int x)` produces the correct answer if it terminates. 
Recall that `x/y` returns the quotient and `x%y` returns the remainder in integer division.

**Requires:** \( x \neq 0 \)

**Returns:** the number of trailing zeros in \( x \). For example, 
\[
\begin{align*}
\text{trailingZeros}(10) &= 1, \\
\text{trailingZeros}(876000) &= 3, \\
\text{trailingZeros}(101101) &= 0.
\end{align*}
\]

```c
int trailingZeros(int x) {
    int zeros = 0;
    while (x%10 == 0) {
        // i%10 gives the remainder mod 10
        // Inv:
        x = x/10;
        zeros = zeros + 1;
    }
    // Postcondition:
    return zeros;
}
```

Hint: Think of the number, e.g., 876000 as \( 876 \times 10^3 \). Write the postcondition and a suitable loop invariant at the designated places. Then argue **partial correctness** as we did in class.
Part II. Specifications

1) Satisfying a specification. Given the specifications and implementations below, enter a check mark in the table if an implementation satisfies the corresponding specification.

Specifications: Assume argument arg is a positive integer

Spec A: “returns: an integer ≥ arg”
Spec B: “returns: a non-negative integer ≥ arg”
Spec C: “returns: arg + 1 ”
Spec D: “returns: arg² “
Spec E: “returns: Integer.MAX_VALUE”

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Implementations:
Code 1: return arg+2;
Code 2: return arg*arg
Code 3: return arg*5;
Code 4: return arg;
Code 5: return Integer.MAX_VALUE;

2) Order the specifications below from weakest to strongest

Spec A: requires: 0 <= arg1 <= 5 & arg2 != 0
returns: 0 <= result <= 3
Spec B: requires: true  // true is the weakest condition
returns: 0 <= result <= 2  if  0 <= arg1 <= 5 & arg2 != 0
result = -1 otherwise
Spec C: requires: 0 <= arg1 <= 5
returns: result = 0 if arg2 != 0
result = -1 otherwise
Spec D: requires: true
returns: false

Answer: __, __, __, __, __
3) Below is the Javadoc specification of a `remove` method from class `ArrayList`.

```java
public E remove(int index)

Removes the element at the specified position in this list. Shifts any subsequent elements to the left (subtracts one from their indices).

Parameters:
    index - index of the element to be removed

Returns:
    The element that was removed from list

Throws:
    `IndexOutOfBoundsException` - if the index is out of range (index < 0 || index > size())

Convert the Javadoc specification into a PoS specification:

**requires:**

**modifies:**

**effects:**

**returns:**

**throws:**

Now, convert your PoS specification into a logical formula:
4) Below is partial implementation of a `add` method in a collection class:

```java
public void add(int index, E element) {
    if (index >= size() || index < 0)
        throw new IndexOutOfBoundsException();
    else
        // add element at index from collection
}
```

For each of the specifications below, circle TRUE if the specification is valid for this code.

a) **TRUE / FALSE**
   
   requires: nothing
   modifies: this
   effects: removes element at index from collection

b) **TRUE / FALSE**
   
   requires: nothing
   modifies: this
   effects: removes element at index from collection
   throws: IndexOutOfBoundsException if index>=size() || index<0

c) **TRUE / FALSE**
   
   requires: index<size() && index>=0
   modifies: this
   effects: removes element at index from collection

d) **TRUE / FALSE**
   
   requires: index<size() && index>=0
   modifies: this
   effects: removes element at index from collection
   throws: IndexOutOfBoundsException if index>=size() || index<0

e) Which of the above do you think is best? Write a brief explanation.
Part III. ADTs

1) Give two advantages of ADTs. Briefly explain your answer.

2)