University of Toronto - Department of Computer Science
CSC410, Fall 2016
Homework 2
Last updated: 4 October 2016

This homework is worth 5% of your final mark

Due: Sunday, 9 October 2016 at 23:59

Instructions:

The homework is to be done individually. You can write the answers on a sheet of paper or using an editor. Handwritten answers must be legible; otherwise, it will affect the mark. Write clearly which problem you are answering.

To submit, either scan your solutions or save your document as a PDF file and upload it to MarkUs.

Note that while all of the material is relevant for the midterm and final, only some problems may be marked.

Questions? Ask them on Piazza (folder hw2).

At the beginning of the homework include and complete the following header :

==========

CSC410, Fall 2016 - Homework 2

Name: _____

Student Number: _____

Lecture: o Monday o Tuesday

I am the sole author of this homework. Signature: _____

==========

Problem 1 (Understanding postconditions) [4 Marks per subproblem]

Can the following methods be proven correct with respect to their specification? If not, explain why and make small changes **to the postcondition** to ensure that they do. (TRUE is not acceptable postcondition)

a)

Precondition: arr != null

```
void changeArray(int arr[]){
```

```
     for (int i=0; i<arr.length/2; ++i){
          arr[i] - -;
     }
}
```
Postcondition: (\forall int i; (i>=0 && i<arr.length) ==> (i < arr.length/2 && arr[i] == \old(arr)[i]-1))

b)
Precondition: arr != null
```
boolean exists(int arr[], int x){
      for (int i=0; i<arr.length; ++i){
            if (arr[i] == x) return true;
      }
      return false;
}
```
Postcondition: !\result <==> (\forall int i; arr[i] != x)

Problem 2 (Preconditions are too weak) [2 Marks per subproblem]

Give a counterexample as to why the following preconditions **are too weak**. (The counterexample should show some input satisfying the precondition such that by the end of the method the postcondition doesn't hold)
a)

Precondition arr!= null
```
int secondElement(int arr[]){
      return arr[1];
}
```
Postcondition: \result == arr[1]

b)

Precondition x >= -10
```
int foo(int x) {
      int y = x* x;
      if (x < 0 ){
            y *= 2;
      }
      if (y < 0){
            y /=2;
      }
      return y;
}
```

Postcondition \result == 200

Problem 3 (finding the weakest precondition - no loops) [5 Marks per subproblem]

Find the weakest precondition. This exercise must be solved using both Hoare triples and the weakest precondition method. **HINT: First find the weakest precondition using the weakest precondition method, then use this result for the Hoare triples method**

a)

```
w = 410;
if (z > 20){
      w += z*5;
}
y += w;
return y;
//@ y>410
```

b)

```
z = 415;
if (x > y){
      x = x-10;
      if (x < (y - 10)){
            z += 10;
      } else {
            z -= 10;
      }
}
//@ z>410
```

Problem 4 (proving correctness of loop invariants)

Prove that the following invariants are preserved by the loop **(with either Hoare triples or the wp method - your choice)**:

```
int powerMinusOne (int y, int n) {
  int x=1;
  int i=0;
  while (i < n) {
     x *= y;
     i++;
  }
```

```
  return x;
}
```

a) `i<=n`

b) `x == pow(y,i)  // where pow(y,i) represents` y^i

Problem 5 (finding the weakest precondition - with loops) [5 Marks per subproblem]

Find the weakest precondition. This exercise must be solved using both Hoare triples and the weakest precondition method.
There is no need to show that the given (partial) loop invariant is preserved.
HINT: You may need to strengthen the loop invariant (and show that the added constraints are preserved).

a)

```
  int i=0;
  int y= 0;
  //@ loop_invariant i <= a.length && a != null && y == (\sum int j; j>=0
& j<i; a[j] )
  while (i < a.length){
    y += a[i];
    i++;
  }
  y += 10;
  //@ y > 420;
```

b)

```
  int x=1;
  int i=0;
  //@ loop_invariant i<=n && x == pow(y,i) // where pow(y,i) represents y^i
  while (i < n) {
     x *= y;
     i++;
  }
 x /= y;
  //@ x <= 100
```