CIS 110 — Introduction to Computer Programming

17 December 2012 — Final Exam

Name: _____

Recitation # (e.g. 201): _____

Pennkey (e.g. bjbrown): _____

My signature below certifies that I have complied with the University of Pennsylvania's Code of Academic Integrity in completing this examination.

_____ _____

Signature Date

Scores:

1		1
2		5
3		15
4		20
5		25
6		30
Total:		96

CIS 110 Exam Instructions

- You have 120 minutes to finish this exam. Time will begin when called by a proctor and end precisely 120 minutes after that time. If you continue writing after the time is called, you will receive a zero for the exam.
- **Make sure your phone is turned off before the exam starts. If it vibrates or rings during the exam, you will receive a substantial penalty.**
- You may not use your phone or open your bag for any reason, including to retrieve or put away pens or pencils, **until you have left the exam room**.
- This exam is *closed-book, closed-notes, and closed-computational devices.* Except where noted, code included in the questions is correct and you may use it as a reference for Java syntax.
- This exam is long. If you get stuck part way through a problem, it may be to your advantage to go on to another problem and come back later if you have time.
- All code must be written out as normal, including all curly braces and semicolons, unless the question states otherwise.
- Do not separate the pages of the exam. If a page becomes loose, write your name on it and use the provided staplers to reattach the sheet when you turn in your exam so that we don't lose it. We reserve the right not to grade any answers on loose sheets of paper.
- Turn in all scratch paper that you use during the exam. Do not take any sheets of paper with you or leave them behind.
- If you require extra paper, please use the backs of the exam pages or the extra sheet(s) of paper provided at the end of the exam. Clearly indicate on the question page where the graders can find the remainder of your work (e.g. "back of page" or "on extra sheet"). Staple an extra sheets you use to the back of your exam when you turn it in using the provided staplers.
- Use a pencil, or blue or black pen to complete the exam. You may use additional colors when writing code if this makes it easier for you, but **do not write in red**. Red is reserved for grading. If you do not have an appropriate writing utensil, raise your hand, and we will give you a pencil.
- If you have any questions, raise your hand and an exam proctor will come to answer them.
- When you turn in your exam, you may be required to show ID. If you forgot to bring your ID, talk to an exam proctor immediately.

Good luck, have fun!

TOY Reference Card

INSTRUCTION FORMATS

```
            | . . . . | . . . . | . . . . | . . . .|
  Format 1: | opcode  |    d    |    s    |    t   |  (0-6, A-B)
  Format 2: | opcode  |    d    |       addr       |  (7-9, C-F)
```

ARITHMETIC and LOGICAL operations

```
    1: add              R[d] <- R[s] +  R[t]
    2: subtract         R[d] <- R[s] -  R[t]
    3: and              R[d] <- R[s] &  R[t]
    4: xor              R[d] <- R[s] ^  R[t]
    5: shift left       R[d] <- R[s] << R[t]
    6: shift right      R[d] <- R[s] >> R[t]
```

TRANSFER between registers and memory

```
    7: load address     R[d] <- addr
    8: load             R[d] <- mem[addr]
    9: store            mem[addr] <- R[d]
    A: load indirect    R[d] <- mem[R[t]]
    B: store indirect   mem[R[t]] <- R[d]
```

CONTROL

```
    0: halt             halt
    C: branch zero      if (R[d] == 0) pc <- addr
    D: branch positive  if (R[d] >  0) pc <- addr
    E: jump register    pc <- R[d]
    F: jump and link    R[d] <- pc; pc <- addr
```

Register 0 always reads 0.
Loads from mem[FF] come from stdin.
Stores to mem[FF] go to stdout.

Miscellaneous

1. (1 points)

   (a) Write your name, recitation number, and PennKey (username) on the front of the exam.

   (b) Sign the certification that you comply with the Penn Academic Integrity Code

True/False

2. (5 points)
   For each question below, circle the correct answer:

   (a) TRUE FALSE For an array `arr`, `arr[length]` will return the last item in the array.

   (b) TRUE FALSE Java classes cannot mix static and non-static methods and variables.

   (c) TRUE FALSE One array cannot contain both `int` and *double* elements.

   (d) TRUE FALSE `(int) (6 * Math.random())` will return a random integer between 1 and 6.

   (e) TRUE FALSE `java printOutput | out.txt` will redirect the output of `printOutput` to the file `out.txt`

   (f) TRUE FALSE The command "`cd ..`" will move to the parent directory in the terminal or command prompt.

   (g) TRUE FALSE `0x001A` in hexadecimal is the same as `00011010` in binary.

   (h) TRUE FALSE In two's complement notation, a binary number has a 1 in the highest-order (most significant) bit if it is positive.

   (i) TRUE FALSE A constructor is the only type of function that can be overloaded.

   (j) TRUE FALSE In a linked list, elements do not have to be sequential in memory.

Babes in TOYland

3. (15 points) The following TOY program does something, but we've forgotten what. All we remember is that the program reads a single value from standard input (at memory address 0x11), and writes a single value to standard output (at memory address 0x21). Answer the questions below, then remind us what the program does. You may assume that the assembly language comments are correct and that the number read from standard input is not negative. Give numeric answers below in ordinary, boring, base 10.

```
01: 0001   (0000 0000 0000 0001,      1)

10: 8501   R[5] <- mem[01]
11: 83FF   read R[3]
12: 1455   R[4] <- R[5] + R[5]
13: 2230   R[2] <- R[3]
14: 1102   R[1] <- R[2]
15: B004   mem[R[4]] <- R[0]
16: C120   if (R[1] == 0) goto 20
17: C21D   if (R[2] == 0) goto 1D
18: AA04   R[A] <- mem[R[4]]
19: 1AA3   R[A] <- R[A] + R[3]
1A: BA04   mem[R[4]] <- R[A]
1B: 2225   R[2] <- R[2] - R[5]
1C: C017   goto 17
1D: 2115   R[1] <- R[1] - R[5]
1E: 1203   R[2] <- R[3]
1F: C016   goto 16
20: 8B02   R[B] <- mem[02]
21: 9BFF   write R[B]
22: 0000   halt
```

(a) What is the value in register `R[4]` at memory address 0x15? _____

(b) What value does the program print if it reads zero from standard input? _____

(c) What value does the program print if it reads one? _____

(d) What value does the program print if it reads five? _____

(e) Describe, **in twenty words or less**, what this program does.

Unbreakable

4. (20 points) For each of the four functions below and on the next page, give an intuitive description **in 20 words or less** of the function's purpose. In addition, **in 30 words or less** state whether the function will compile and run properly in all cases and what the error(s) will be otherwise. Circle your answer.

(a)
```
public static void one(int x) {
  while (x != 0) {
    System.out.println(x);
    x--;
  }
}
```

(b)
```
public static int two(char x) {
  String keyboard = "abcdefghijklmnopqrstuvwxyz";
  for (int i = 0; i < keyboard.length(); i++)
    if (keyboard.charAt(i) == x)
      return i;
}
```

(c)
```
public static int three(int[] x) {
  if (x == null) return 0;

  int sum = 0;
  for (int i = 0; i < x.length; i++)
    sum += x[i];
  return sum / x.length;
}
```

(d)
```
public static void four(String[] x) {
  for (int i = 0; i < x.length; i++) {
    x[i] = "" + x[i].length();
    System.out.println(x[i]);
  }
}
```

Something or Other

5. (25 points) Consider the following program, then answer the questions on the next page:

```
public class Other {
    public static int foo = 0;
}

public class Something {
  private Something s1;
  private Something s2;
  private int x;

  public Something (int x) {
    if (x < 1) {
      s1 = null;
      s2 = null;
    } else {
      s1 = new Something(x / 2);
      s2 = new Something(x / 2);
    }

    this.x = x;
    Other.foo++;
  }

  public int sum() {
    if (s1 == null || s2 == null) return x;
    else return x + s1.sum() + s2.sum();
  }

  public static void main(String[] args) {
    Something s = new Something(Integer.parseInt(args[0]));
    System.out.println("sum: " + s.sum());
    System.out.println("foo: " + Other.foo);
  }
}
```

What does the program `Something` print when run with the following arguments?

(a) `% java Something 0`

(b) `% java Something 1`

(c) `% java Something 2`

(d) `% java Something 3`

(e) `% java Something 4`

Know Your Node Code

6. (30 points) Consider a linked list of integer values that uses the linked list node type:

```
public class Node {
  public Node next;
  public int  value;
}
```

Write a `public static` function `remove` that takes a linked list of `Node`s `list` and an integer `val` as arguments, modifies `list` by removing all nodes with the value `val`, and returns the modified list. You only need to write the `remove()` function, not the surrounding class. (Hint 1: You do not need to create any new nodes with the `new` operator. Hint 2: Don't look for a clever solution, just write something that works.)

Postscript (extra paper)

Postscript (extra paper)