

CIS 110 — Introduction to Computer Programming

29 April 2013 — 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:

0		1
1		10
2		12
3		20
4		25
5		30
Total:		98

### 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.**
- Food and drink are strictly forbidden, **including water and gum.**
- 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.
- 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. All other colors are 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 and 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	$R[d] \leftarrow R[s] + R[t]$
2: subtract	$R[d] \leftarrow R[s] - R[t]$
3: and	$R[d] \leftarrow R[s] \& R[t]$
4: xor	$R[d] \leftarrow R[s] \wedge R[t]$
5: shift left	$R[d] \leftarrow R[s] \ll R[t]$
6: shift right	$R[d] \leftarrow R[s] \gg R[t]$

## TRANSFER between registers and memory

7: load address	$R[d] \leftarrow \text{addr}$
8: load	$R[d] \leftarrow \text{mem}[\text{addr}]$
9: store	$\text{mem}[\text{addr}] \leftarrow R[d]$
A: load indirect	$R[d] \leftarrow \text{mem}[R[t]]$
B: store indirect	$\text{mem}[R[t]] \leftarrow R[d]$

## CONTROL

0: halt	halt
C: branch zero	if ( $R[d] == 0$ ) $pc \leftarrow \text{addr}$
D: branch positive	if ( $R[d] > 0$ ) $pc \leftarrow \text{addr}$
E: jump register	$pc \leftarrow R[d]$
F: jump and link	$R[d] \leftarrow pc; pc \leftarrow \text{addr}$

Register 0 always reads 0.

Loads from mem[FF] come from stdin.

Stores to mem[FF] go to stdout.

**Miscellaneous**

0. (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

**Multiple Choice (2 Pages)**

1. (10 points) For each of the following questions, circle the correct answer:

- (a) What happens if you try to access a private variable from outside the class in which it is declared?
  - i. The program runs normally.
  - ii. The program runs, but treats the variable as having the value 0 or `null`.
  - iii. The program runs, but the statement accessing the private variable is ignored.
  - iv. The program runs, but behaves unpredictably.
  - v. The program compiles, but a run-time error occurs.
  - vi. The program does not compile.
  
- (b) `5 / 2 * Math.random()` generates a random number between...
  - i. 0 and 2.5.
  - ii. 0 and 2.
  - iii. 2 and 5.
  - iv. 2.5 and 5.
  - v. 2 and  $\infty$ .
  - vi. 2.5 and  $\infty$ .

**Multiple Choice (Cont'd)**

(c) What does the following function return?

```
public static boolean foo(boolean x, boolean y, boolean z) {  
    if (x && y) return !x || z;  
    else      return (x && y) || z;  
}
```

- i. x
- ii. y
- iii. z
- iv. x && y
- v. true
- vi. true if either x and y are both true or z is true, and false otherwise

(d) The `toString()` method of an object should do which of the following?

- i. Print the object to standard output in a sensible way.
- ii. Return a sensible `String` representation of the object.
- iii. Print the object to standard output in a sensible way and return that output as a `String`.
- iv. Whatever the object's designer wants it to do.
- v. Read in the contents of the object from standard input.
- vi. None of the above

(e) Which of the following expressions computes the length of a string `s`?

- i. `s.length`
- ii. `s.length()`
- iii. `String.s.length`
- iv. `String.s.length()`
- v. `String.length`
- vi. `String.length()`

**Easy as ABC (2 Pages)**

2. (12 points)

Consider the code below, then answer the questions on the following page.

```
public class A {
    private B b;
    public String c;
    public A(B b) {
        b = b;
    }

    public A(B d, String e) {
        b = d;
        c = e;
    }

    public String get() {
        return b.c + " " + c;
    }
}

public class B {
    public static int a;
    public String c;
    public B(int b, String d) {
        a = b;
        c = d;
    }
}

public class C {
    public B[] b;
    public String c;
    public C() {
        b = new B[2];
        c = "f";
        b[0].a = 2;
        b[1].a = 4;
        b[0].c = "a";
        b[1].c = "d";
    }
    public C(B a) {
        b = new B[1];
        b[0] = a;
    }
}
```

**Easy as ABC... (Cont'd)**

For each question below, what will the code snippet print out? If there is an error, your answer should show that you know what causes the error, but you do not need to reproduce Java's error message exactly. Put a box around your answer so we can distinguish it from your scratch work.

(a) 

```
B b = new B(7, "bad");
A a = new A(b);
System.out.println(a.get());
```

(b) 

```
B b = new B(7, "bad");
A a = new A(b, b.c);
System.out.println(a.get());
```

(c) 

```
B b = new B(7, "bad");
A a = new A(b, b.c);
System.out.println(a.c);
```

(d) 

```
C b = new C();
System.out.println(b.c);
```

**Toying With TOY (2 Pages)**

3. (20 points)

Consider the following TOY program, then answer the questions on the following page. You may assume that all assembly comments are correct.

```
10: 81FF  read R[1]
11: 82FF  read R[2]
12: 7400  R[4] <- 0000
13: 7501  R[5] <- 0001
14: 7602  R[6] <- 0002
15: 2321  R[3] <- R[2] - R[1]
16: D31B  if (R[3] > 0) goto 1B
17: 1445  R[4] <- R[4] + R[5]
18: C31C  if (R[3] == 0) goto 1C
19: 2112  R[1] <- R[1] - R[2]
1A: C015  if (R[0] == 0) goto 15
1B: 2323  R[3] <- R[2] - R[3]
1C: 94FF  write R[4]
1D: C622  if (R[6] == 0) goto 22
1E: 1444  R[4] <- R[4] + R[4]
1F: 2665  R[6] <- R[6] - R[5]
20: 1333  R[3] <- R[3] + R[3]
21: C01D  if (R[0] == 0) goto 1D
22: C226  if (R[2] == 0) goto 26
23: 1646  R[6] <- R[4] + R[6]
24: 2225  R[2] <- R[2] - R[5]
25: C022  if (R[0] == 0) goto 22
26: 1563  R[5] <- R[6] + R[3]
27: 95FF  write R[5]
28: 0000  halt
```

**Toying With TOY (Cont'd)**

(a) For each pair of numbers below, what will the program write to standard output? Write your answers in decimal (base 10) and put a box around it.

i. 1 and 1

ii. 2 and 1

iii. 1 and 2

iv. 0 and 1

v. 0 and 0

vi. 1 and 0

(b) In simple formulas, what values are outputted if the first input is **larger** than the second?

(c) In simple formulas, what values are outputted if the first input is **smaller** than the second?

**The Tree of Knowledge (2 Pages)**

4. (25 points)

Consider the linked data structure below, then answer the questions on the following page.

```
public class Knowledge {
    private int apple;
    Knowledge snake, eve;

    public void insert(int orange) {
        Knowledge adam = new Knowledge();
        adam.apple = orange;
        insert(adam);
    }

    private void insert(Knowledge adam) {
        if (adam.apple < apple) {
            if (eve != null) eve.insert(adam);
            else
                eve = adam;
        } else {
            if (snake != null) snake.insert(adam);
            else
                snake = adam;
        }
    }

    public void A() {
        if (snake != null) snake.A();
        System.out.print(apple + " ");
        if (eve != null) eve.A();
    }

    public void B() {
        System.out.print(apple + " ");
        if (snake != null) snake.B();
        if (eve != null) eve.B();
    }

    public void C() {
        if (eve != null) eve.C();
        if (snake != null) snake.C();
        System.out.print(apple + " ");
    }
}
```

**Tree of Knowledge (Cont'd)**

(a) If the following code is executed:

```
int[] vals = { 3, 7, 8, 2, 9, 1, 6, 4 };
Knowledge k = new Knowledge();
k.apple = 5;
for (int i = 0; i < vals.length; i++) k.insert(vals[i]);
```

i. What will `k.A()` print?

ii. What will `k.B()` print?

iii. What will `k.C()` print?

(b) For each of the outputs below, say which of `A()`, `B()`, and/or `C()` could have produced it. For each answer, give an order in which you could insert the elements to produce the output. Write “none” if none of them could have produced it. Put a box around your answer so we can distinguish it from your scratch work.

Assume you create a single `Knowledge` object, then progressively add values using the `insert()` method, just like the code snippet in part (a).

i. 1 2 3 4 5

ii. 5 4 3 2 1

iii. 2 3 4 3

**Leave and Let Leave (2 Pages)**

5. (30 points)

Write a static function `interleave` that interleaves two linked lists, `a` and `b`, and returns the result. The first node of the interleaved list should be the first node of `a`, followed by the first node of `b`, then the second node of `a`, the second node of `b`, and so on. After the shorter list's nodes are exhausted, append all remaining nodes from the longer list to the end.

Assume the following `Node` class:

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

Actually, write two versions of `interleave`, a recursive version and an iterative version. You should not use the `new` keyword in either version. The only assumption you may make about `a` and `b` is that they are not circular.

*(Hints: The iterative version is more code, but the recursive version is more subtle. Start with whichever version seems more natural to you. In both version, start by assuming both lists are the same length; add code to deal with special cases if you have time.)*

- (a) Write a recursive version of `interleave()`. This version may not contain any `for` or `while` loops. For full credit, `interleave()` should call itself recursively, but must not call any other functions. However, we will accept answers where `interleave()` calls a single, recursive helper function that you write as part of your answer. You may not use more than one helper function, or call any other functions. If you write a helper function, it can take any arguments you wish. You do not need to write a class around your function(s), and you do not need to write any comments.

**Leave and Let Leave (Cont'd)**

- (b) Write an iterative version of `interleave()`. This version may not call any other functions, and you may not write a separate, helper function. You may use `for` or `while` loops. You do not need to write a class around your function, and you do not need to write any comments.

**Postscript (extra paper)**

**Postscript (extra paper)**