Name_____

NetID_____

Prelim One Solution

CS211 — Fall 2005

Closed book; closed notes; no calculators. Write your name and netID above. Write your name clearly on ***each*** page of this exam. For partial credit, you must show your work.

Do not begin until you are instructed to do so. You have 90 minutes.

Name_____

1. (*20 points; 2 each*) True or false?

a) T **F** The statement "`String s;`" calls the constructor of class Object.

b) **T** F The `protected` modifier signifies that the method or variable can only be accessed from within its class, from within its subclasses, or from within classes in the same package.

c) T **F** In Java, a class may directly inherit from more than one abstract class.

d) **T** F If class A inherits from abstract class B, and B implements an interface C, then a cast from an object of type A to an object of type C will always succeed. Example:

```
A a;
...
C c = (C) a;
```

e) **T** F In Java, an interface can extend multiple interfaces.

f) T **F** Strong induction is much better than weak induction, and therefore we should use it whenever possible.

g) T **F** In Java, objects are stored in the stack frames.

h) **T** F If a variable is declared to hold a primitive type then its static and dynamic types are always the same.

i) **T** F During a Java program's execution, a single method can correspond to multiple stack frames.

j) **T** F Any binary tree of depth 2 has ≤ 7 nodes.

Name_____

2. (*Induction; 20 points; 10 each*)

Make sure you clearly show (i) your base case or cases, (ii) your induction hypothesis, (iii) the inductive step, and (iv) your conclusion.

2a. Use induction to prove that any amount of postage worth 12 cents or more can be formed using just 4-cent and 5-cent stamps. In other words, show P(n) holds for all $n \geq 12$ where P(n) represents: "Postage of n cents can be formed using 4-cent and 5-cent stamps."

Base Case: 12 = 4 + 4 + 4

Induction Hypothesis: P(k) holds for some k >= 12

Inductive Step: Consider the solution for k. Either
(a) the solution includes a 4 cent stamp
In this case we replace the 4 with a 5 to get a solution for k+1
or (b) the solution consists entirely of 5 cent stamps
In this case, we replace 3 5-cent stamps with 4 4-cent stamps to make a solution for k+1. We know there are 3 5-cent stamps because we are only considering values >= 12.

Conclusion: P(n) holds for all n >= 12.

2b. Use induction to prove that 10^n has a remainder of 1 after dividing by 9 for all $n \geq 1$. (Hint: an integer that has a remainder of 1 after dividing by 9 can be written as 9k+1 for some integer k.)

Base Case: 10^1 has remainder 1 when divided by 9.

Induction Hypothesis: 10^n has remainder 1 when divided by 9 for some n >= 1.

Inductive Step: Consider $10^{n+1} = 10\ (10^n)$. By the IH and the hint, this can be written as 10 (9k + 1) for some integer k. Rewriting, this is 9(10k + 1) + 1. It's easy to see that this has remainder 1 when divided by 9.

Conclusion: 10^n has remainder 1 when divided by 9 for all n >= 1.

Name_____

3. (*Recursion; 10 points*)
Write a *recursive* method (called removeEven) to remove the elements at even positions in a linked list. Assume that your method takes a single argument of type ListCell and that it returns a value of type ListCell; the declaration for ListCell is given below.

```
class ListCell {
    public Object datum;     // Data for this cell
    public ListCell next;    // Next cell
}
```

```
ListCell removeEven (ListCell list) {
        if (list == null) return null;
        if (list.next == null) return null;
        list.next.next = removeEven(list.next.next)
        return list.next;
    }
```

Name_____

*4. (*Types; 24 points; 3 each*)*

Consider the following interface and class declarations.

```
interface I1 {...}
interface I2 {...}
interface I3 extends I1 {...}
class C1 implements I1 {...}
class C2 implements I2 {...}
class C3 implements I3 {...}
class C4 extends C3 implements I2 {...}
```

For each of the following snippets of Java code, determine whether the code will produce no error, a run-time error, or a compile-time error. (Assume that each snippet is tested independently of the others.)

a) `I2 a = new I2();` — *Compile Error*

b) `I2 b = new C2();` — *OK*

c) `C3 c = new C4();` — *OK*

d) `C2 d = new C4();` — *Compile Error*

e) `C4 e = new C3();` — *Compile Error*

f) `C4 f = (C4)(new C3());` — *Runtime Error*

g) — *OK*

```
I1 g1 = new C1();
C4 g2 = new C4();
g1 = g2;
```

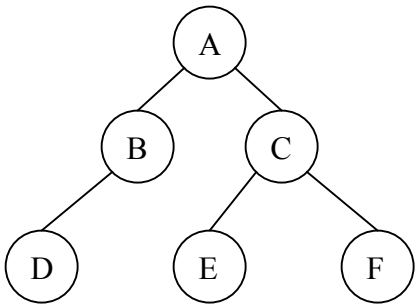
h) — *Compile Error*

```
I1 g1 = new C4();
I2 g2 = new C2();
g2 = g1;
```

Name_____

5. (*Trees; 21 points; 7 each*)

5a. In what order are nodes processed during preorder, inorder, and postorder traversals of the following binary tree.

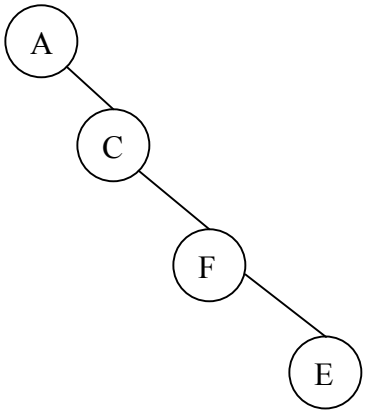


preorder: *ABDCEF*

inorder: *DBAECF*

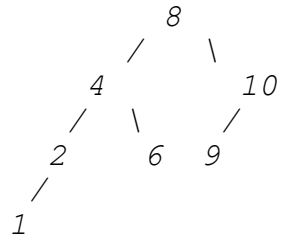
postorder: *DBEFCA*

5b. Draw a binary tree (with at least 4 nodes) that processes nodes in the same order for both its preorder and its inorder traversals, but processes nodes in a different order for its postorder traversal.



5c. If we construct a Binary Search Tree (BST) by inserting elements 8, 4, 6, 2, 1, 10, 9 in the given order then what is the sum of the elements that are stored the leaf nodes of the resulting tree? (It is possible to receive partial credit, but only if you show your work.)

Sum = 16



Name_____

6. (*Recursion; 5 points*)
Write Java code for a class MainProblem that contains just a single method: the main method. When run using the command line with one argument, MainProblem should print the argument followed by the argument without its first character, followed by the argument without its next character, etc. Example:

```
> java MainProblem hello
```

Output: `hello ello llo lo o`

Specifications:

- Assume legal input.
- Do not use any loop control-structures (i.e., no for-loops, no while-loops, etc.).
- There should be no methods other than the main method.
- Do not use any class variables.

Hint: You are likely to find the substring method of class String to be useful. Here is its documentation from the Java API.

String substring(int beginIndex)
Returns a new string that is a substring of this string.
String substring(int beginIndex, int endIndex)
Returns a new string that is a substring of this string.

```
public class MainProblem {
   public static void main (String[] args) {
      if (args[0].length() == 0)
         System.out.println();
      else {
         System.out.print(args[0] + " ");
         args[0] = args[0].substring(1);
         main(args);
      }
   }
}
```

Name_____

Bonus Point Questions

Work on these *only if* you have extra time. Bonus points are not used in determining your final course grade unless you are on a grading borderline.

B1. What is the answer to this question?
???

B2.
Consider the following two classes. What is printed when the main method is executed? (You will probably need to mark off lengths on a straightedge to distinguish between variables. You can tear off a scrap of paper from the last page for this purpose, but we still want your course feedback and the feedback is worth 5 bonus points.)

```
  public class __________ {
    static int __________;
    public static void main(String[] ____________) {
      ________ _________ = new _________( (new
_________(++__________.__________)).__________ );
      _________(_________._________(_________._________));
      _________(_________._________(_________)._________);
      _________(_________(_________._________(_________)));
    } // method main
    static int _________(________ _________) {return _________._________;}
    public static void _________(int _________)
{System.out.println(_________);}
  } // class __________


  class _________ {
    int _________;
    _________(int _________) {
      this(_________,_________++);
      this._________=_________;
    }
    _________(int ________, int _________) {
      this._________+=_________+=this._________+________++;
    }
    int _________(int _________) {
      _________(this);
      return _________;
    }
    ________ _________(________ _________) {
      _________._________+=++__________.__________;
      return new _________(_________._________);
    }
  } // class _________
```

Prints: 3 9 13 (on separate lines)

Name_____No name needed here_____

B3. Course Feedback

You can detach this sheet from the exam and hand it to a TA to maintain confidentiality. If you turn in this Course Feedback form, you receive 5 bonus points.

1. How many hours per week are you spending on the homework for this course?

2. Do you attend lecture regularly? What can we do to improve the lectures?

3. Do you attend section regularly? What can we do to improve section?

4. Do you see the consultants for assistance? What can we do to improve consulting?