

CS 61B, Summer 99**Midterm 1****1.0 Putting your login on every page (1pt, 3mins)**

Write your login using CAPITAL LETTERS on the upper left-hand corner of every page.

2.0 Short Answer (3pts, 10mins)

The following section contains several short questions. Answer them as concisely as you can.

Why doesn't this compile and how can you fix it (make one change only)?

```
public class Something {
    static int a;

    public static void foo() {
        a = 5;
    }
}
```

Something.java:5: Can't make a static reference to nonstatic variable a in class Something.

```
a = 5;
```

Can a private instance variable belonging to the current object ever be accessible by another object that is not the current object. If so, describe a situation when this can occur.

What effect does the public and private modifiers have when placed in front of methods.

3.0 Object Oriented Design (4pts, 10mins)

Han has designed two classes for the purposes of the first project, Board and MineSweeper. The Board class uses two 2-d arrays to maintain the state of the game. These two arrays are:

`covered` - keeps track of which cells are covered, which cells are uncovered, and which cells have been marked by the user.

`mineField` - a representation of the actual board-state. If a cell is a mine, its value is -1 (or Board.BOMB), otherwise its value is equal to the number of mines surrounding it.

The MineSweeper class collects initial information from the user about a new game and then starts the game using the Board class to maintain state. To allow the MineSweeper class to access information about `covered` and `mineField` (so that MineSweeper may print out the state of the board), Han creates the following accessor functions:

```
public int[][] covered () {
    return covered;
}

public int[][] mineField() {
    return mineField;
}
```

Although this technique works, it is a bad idea from a class design standpoint. Explain why providing access to the state of the Board class in this fashion is a bad idea, and explain what you would have done instead so that the MineSweeper class could have access to the information inside the `covered` and `mineField` arrays. Provide your answer on the next page.

4.0 Linked Lists (8pts, 30 mins)

Below is a partial excerpt of the singly linked list implementation Susan Lee presented in lecture. You will need to use these classes to implement an additional method in the `IntList` class. See the problem description below the provided code.

```
public class IntNode {
    private int value;    //the number this holds
    private IntNode next; //reference to next node

public IntNode(int number) {
    value = number;
    next = null;
}

public IntNode(int number, IntNode node){
    value = number;
    next = node;
}

public int getValue() {
    return value;
}

public void setValue(int newValue) {
    value = newValue;
}

public IntNode getNext() {
    return next;
}

public void setNext (IntNode node) {
    next = node;
} // IntNode

public class IntList {
    private IntNode front; // the first node in the list

public IntList() {
    front = null;
}

public IntList(IntNode node) {
```

```

    front = node;
}
...
} // IntList

```

Create an instance method of the IntList class called reverse() which reverses the order of the nodes referred to by the IntNode variable front. You may **NOT** use the new operator. Any solution that does will be marked wrong. After a call to this method, the variable front should refer to what used to be the last node in the list, its next variable should refer to what used to be the second to the last variable, etc. You may add additional private variables/methods if you feel it is necessary. Provide your answer on the following page.

Answer to question 5.0:

```
public void reverse() {
```

5.0 Flow of Control (4pts, 10mins)

One "brute force" way of testing whether or not a number is prime is to start from the first integer less than or equal to the square root of the number to be tested and to check and make sure all such numbers between this number and 2 do not divide evenly into the candidate prime number.

For example, to check if 173 was prime, one would check if 13, 12, 11, 10,9...2 divided evenly into 173. If any of them do, the number is not prime.

Write a public static method called boolean isPrime (int candidate) that returns true when the candidate is prime. Assume that the candidate integer is greater than 0.

The following method in the java.lang.Math class might be useful. The term NaN stands for "Not a Number." To convert a double value to an int, the following code fragment might be helpful
int something = (int) 3.14;

```
public static double sqrt (double a)
```

Returns the square root of a double value.

Parameters:

a - a double value.

Returns:

the square root of a. If the argument is NaN or less than zero, the result is NaN.

Answer to question 5.0:

```
public static boolean isPrime (int candidate) {
```

6.0 True or False (4pts, 10 mins)

Put a "T" or an "F" next to each statement.

1. The use of the "this" keyword is often optional.
2. Instance methods have access to all static variables that are part of its enclosing class regardless of access modifier.
3. Local variables of methods are not allowed to possess the same names as instance or class variables in the same class (the compiler will complain).
4. Class and instance variables must have a public or private access modifier.
5. All classes must have a main method.
6. Static methods must always be called with the name of the class it is associated with.
7. If it is possible to create an instance of a class, then the programmer must have defined a constructor for that class.
8. If the dot operator is applied to a reference variable that stores null, nothing happens.

**Posted by HKN (Electrical Engineering and Computer Science Honor Society)
University of California at Berkeley**
**If you have any questions about these online exams
please contact <mailto:examfile@hkn.eecs.berkeley.edu>**