

-----  
MT 1 Solutions

Each major numbered question is equally weighted.  
Within a multipart question, each part is worth the same amount.  
For questions 1-4, each component was graded on the following scale:  
2 - clearly demonstrates understanding of the question being asked  
1 - demonstrates some understanding  
0 - demonstrates no understanding

No fractional points were assigned.

For question 5, grades were assigned on a four point scale at half point increments.

- 4 - demonstrates complete understanding
- 3 - demonstrates general understanding, missing at most one concept
- 2 - demonstrates good understanding, missing no more than a few concepts.
- 1 - demonstrates weak understanding, missing many concepts.
- 0 - demonstrates no understanding

Your ultimate score was determined by taking your raw scores and scaling such that a perfect score would be worth 100 points (5 questions at 20 points each). so, for example, if you received a raw score of 3 on question 5, your scaled score would be 15.

If you would like to challenge your score, please read the following carefully:

1. If you believe that we have added or scaled improperly, please PROMPTLY bring the exam to your TA and explain the miscalculation.
2. If you believe that we have graded you improperly in light of the solutions, please write a short description of why you think your answer should be re-evaluated and submit, stapled to the back of your exam, to your TA by Tuesday February 17th in section.

-----  
Question 1. Consider this illegal class definition:

A. Why must it be the case that the first constructor is illegal? (that is, don't recite a rule, but explain why a rule exists).

Two possible answers:

1. The call to super must be the first line of code to ensure that the base class is properly constructed prior to any execution of the subclass's code. The second call to super violates this rule.

OR

2. An object can be constructed only once. It makes no sense to call the base class' constructor a second time.

B. Similarly, explain why the second constructor must be illegal.

See <1> from part a.

□

-----  
Question 2. Consider this illegal class definition:

```
class Sample {
    Integer i;
    public Sample() { i = new Integer(100); }
    private Object get()
        { return i; }
    private void set(Object o)
        { i = o; }
}
```

A. Why does this code not compile?

It is illegal to assign an instance of a base class to an instance of its subclass. The assignment `i=o` in `set` is illegal.

B. How would you fix the code so that it does compile?

Various options:

1. Change the signature for `set` to `set(Integer o)`
2. Change the type of `i` to `Object i`
3. Cast the assignment to `i` as `i = (Integer)o`

□

-----  
Question 3. Stacks of Stacks

A. What problem is solved by the stack?

How do I retrieve a set of stored items in Last In First Out order.

B. The javadoc comments from the `InStack` interface below have gone missing. Using appropriate javadoc syntax and style, put them back in the space provided.

```
public interface IntStack {
    /**
     * Stores the specified item in LIFO order.
     * @param item the item to be stored.
```

```
*/  
void push(int item);  
  
/**  
 * Retrieves that most recently stored item.  
 * @return most recently stored item  
 */  
int pop();  
  
}
```

C. a. Under what circumstances can push fail?  
Stack is full (OVERFLOW)

b. Under what circumstances can pop fail?  
Stack is empty (UNDERFLOW)

D. The FancyStack class given below (and in class and on the web site) has a serious bug.

Describe the bug.

Sense of isEmpty is reversed. Returns true when stack is not empty, and false when it is empty.

Show below how to fix the bug.

change isEmpty to:

```
public boolean isEmpty()  
{  
    return (curSize == nFree);  
}
```

---

□  
Question 4. Drivers

A. List two important problems solved by the DriverHelper class? (that is, state which problems are solved by the class as a whole, not which problems are solved by each individual method).

1. How do I....

get out of static land.

2. How do I....

avoid having to write the same generic program initialization and argument parsing code over and over again.

B. Why are several of the methods defined as abstract?

Because they are not defined in DriverHelper. They are to be defined by the subclass of DriverHelper.

C. Why is the class as a whole defined as abstract?

Any class with abstract methods must itself be abstract.

D. What limitation is imposed on an abstract class?

It is not instantiable ("newable").

---

□  
Question 5: A Command Line Calculator

Calculator.java

---

## CSE143\_mt1\_solutions

```
/**
 * Driver main module.
 */
public class Calculator
{
    /**
     * Main routine.
     * args -- Items to be pushed onto the stack and then popped off.
     Simple demo.
     */
    static public void main (String args[]) {
        CalculatorHelper ch = new CalculatorHelper("Calculator");
        ch.go(args);
    }
}
```

### CalculatorHelper.java

---

```
public class CalculatorHelper extends DriverHelper {
    public CalculatorHelper(String name)
    {
        super(name);
    }

    protected void defineArguments()
    {
        arguments.expectUnflaggedArguments("int");
        arguments.addCommand("-add", "sum arguments");
        arguments.addCommand("-mul", "mul arguments");
    }

    protected void initialize()
    {
    }

    protected void process()
    {
        String[] nums = arguments.getNonFlaggedArguments();
        boolean add = false;

        if (arguments.isFlagPresent("-add"))
            add = true;
        else if (arguments.isFlagPresent("-mul"))
            add = false;
        else
            exit(-1, "Neither -add nor -mul present");

        int result = 1;
        if (nums.length == 0 || add)
            result = 0;

        for (int i = 0; i < nums.length; i++) {
            int k = Integer.parseInt(nums[i]);
        }
    }
}
```

```
                                CSE143_mt1_solutions
    if (add)
        result = result + k;
    else
        result = result * k;
    }
    System.out.println("Result is " + result);
}

}
```

-----

Primarily, I was looking for you to be able to define the program in terms of the code we've discussed in class, provided on the web site, and included in the exam.