CSC 108H1 S February Midterm 2004
Duration — 50 minutes
Aids allowed: none

Student Number: |_|_|_|_|_|_|_|_|_|

Lab day, time, room: ____________

Last Name: ____________ **First Name:** ____________

Lecture Section: ____________ **Instructor:** ____________

Do ***not*** *turn this page until you have received the signal to start.*
(Please fill out the identification section above,
and read the instructions below.) *Good Luck!*

This midterm consists of 3 questions on 5 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.*
Comments are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code.
For 1 bonus mark write your student number at the bottom of pages 2-5 of this test.
If you use any space for rough work, indicate clearly what you want marked.

# 1:	______/14
# 2:	______/18
# 3:	______/18
TOTAL:	______/50

Short Java API descriptions (all methods are public):

```
class Integer:
  Integer(int i) // An Integer with value i
  static int parseInt(String s) // = s's value, as an int.
class Double:
  Double(double d) // A Double with value d
  static double parseDouble(String s) // = s's value, as a double.
class String:
  String substring(int i, int j) // = the letters between i (inclusive) and j (non-inclusive).
  String substring(int i) // = the letters from i (inclusive) to the end.
  int indexOf(String s) // = the index of s in this String; -1 if s is not a substring.
  int indexOf(String s, int i) // = index of s in this String after index i; -1 if s not found.
  int length() // = the number of characters in this String.
class JOptionPane:
  static String showInputDialog(String m) // get input from the user, prompting with m.
```

Question 1. [14 MARKS]

1. Given two `double` variables `balance` and `interest`, and an `int` variable `amount`, write an expression that multiplies `balance` by `interest`, and stores the result in `amount`.

2. Given an `int` variable `total`, write an expression that evaluates to true if the value of `total` is even, and evaluates to false otherwise.

3. Given `double` variables `cash` and `bill`, write an expression that evaluates to true if `cash` is negative or if `cash` is greater than `bill`, and evaluates to false otherwise.

4. Write a statement that declares a `Double` reference named `grade`, and creates and assigns it an instance of `Double` with an initial value of 75.5.

5. Write the definition of a class `Computer`. The class has no methods, one instance variable of type `String` called `ipAddress`, and one static `int` variable called `quantity`, which is initialized to `10`.

6. Write the definition of the method `makeAddress`, which has an `int` parameter representing the street number and a `String` parameter representing the street name. The method returns a `String`. For example, given the parameters 111 and Main Street, the method would return a `String` of the form: I live at 111 Main Street

7. You are given a class called `Phone`, which has an instance variable of type `String` called `number` and a `boolean` instance variable called `broken`. Write a constructor for the class that takes a `Phone` object as a parameter and copies that object's instance variable values to the object being created.

Question 2. [18 MARKS]

In lecture, we built a Java class used to keep track of books in a library. It looked something like this:

```
class BookRecord {
   private String ISBN;
   private String author;
   private String title;

   public void setISBN(String i) { this.ISBN = i; }
   public void setAuthor(String a) { this.author = a; }
   public void setTitle(String t) { this.title = t; }

   public String getISBN() { return this.ISBN; }
   public String getAuthor() { return this.author; }
   public String getTitle() { return this.title; }

   public String toString() {
      return this.ISBN + " :  " + this.author + " :  " + this.title;
   }
}
```

Now, you will write a new class, `SalesBookRecord`, which extends the `BookRecord` class for use in a system that is used to sell books. Specifically, your new class will include the following:

1. a private field used to store the price of the book (in dollars, allowing for prices such as 10.95).
2. public get and set methods for price.
3. A `discount` function, which takes a discount, n, (int between 0 and 100), and returns the price of the book discounted by n percent (for example, if the price of the book is 10.0, and this method is called with a value of 30, the function returns 7.0, but leaves the price of the book unchanged).
4. A `toString` function, which returns a `String` representation of this `SalesBookRecord` in the following format:

   author : title : price : isbn

   Write your class on the following page. Be sure to include some comments for tricky parts of code.

Write your answer to Question 2 here:

Question 3. [18 MARKS]

Consider these two classes.

```
public class Bill {                         |  public class Schedule {
                                            |
  private static int total = 0;             |    private String day;
  private String utility;                   |    private Bill bill;
  private int amount;                       |
  private boolean paid = false;             |    public Schedule(String d) {
                                            |      this.day = d;
  public Bill(String u, int a) {            |    }
    this.utility = u;                       |
    this.amount = a;                        |    public void setBill(Bill b) {
    Bill.total = Bill.total + this.amount;  |      this.bill = b;
  }                                         |    }
                                            |
  public void pay() {                       |    public Bill getBill() {
    this.paid = true;                       |      return this.bill;
  }                                         |    }
                                            |
  public String toString() {                |    public String getDay() {
    return this.utility                     |
             + ", paid " + this.paid;       |      return this.day;
  }                                         |    }
                                            |
  public static int getTotal() {            |    public String toString() {
    return Bill.total;                      |      return this.day + ", "
  }                                         |             + getBill();
}                                           |    }
                                            |  }
```

On the next page is a `TestCase` subclass that tests these classes. There are three `test` methods and 7 `assertEquals` calls. In the space provided, for each `assertEquals` call write:

P if the `assertEquals` passes,

F if the `assertEquals` fails, and

N if the `assertEquals` call is not reached because a previous `assertEquals` failed.

To the right of each failure, write the actual value of the 2nd parameter in the space provided.

```
                                                      Result    If failed,
                                                      (P/F/N)   actual value
                                                                of 2nd parameter?
import junit.framework.TestCase;
public class BillTester extends TestCase {

  public void test1() {
    Bill b = new Bill("gas", 1255);
    assertEquals("gas, paid false", b.toString());    ----    --------------------
    b.pay();
    assertEquals(1255, Bill.getTotal());               ----    --------------------

    assertEquals("gas, paid true", b.toString());     ----    --------------------
  }

 public void test2() {
    Bill b1 = new Bill("gas", 1255);
    Bill b2 = new Bill("electricity", 550);
    Schedule s = new Schedule("Monday");
    s.setBill(b1);
    b1 = b2;
    s.getBill().pay();
    assertEquals("Monday, gas, paid true",            ----    --------------------
                  s.toString());
  }

  public void test3() {
    Schedule s1 = new Schedule("Tuesday");
    Schedule s2 = new Schedule("Thursday");
    Bill b1 = new Bill("gas", 1255);
    Bill b2 = new Bill("electricity", 550);
    s1.setBill(b1);
    s2.setBill(b2);
    s2 = s1;
    s1.getBill().pay();
    assertEquals("Tuesday", s2.getDay());              ----    --------------------

    assertEquals(1805, Bill.getTotal());               ----    --------------------

    assertEquals("electricity, paid true",             ----    --------------------
                 s1.getBill().toString());
  }
```

```
                                                                 Result    If failed,
                                                                 (P/F/N)   actual value
                                                                           of 2nd parameter?

  public void test4() {
    Schedule s1 = new Schedule("Monday");
    Schedule s2 = new Schedule("Friday");
    Bill b = new Bill("electricity", 315);
    s1.setBill(b);
    b.pay();
    s2.setBill(b);
    assertEquals(s1.getDay().substring(3),                       ----    --------------------
                 s2.getDay().substring(3));

    assertEquals(s1.getBill().toString(),                        ----    --------------------
                 s2.getBill().toString());
  }

}
```

Use this page for rough work.

You can tear this page off, but we will collect it. You must fill in your student number if you tear it off, and you will lose 20% if you keep this page.

Total Marks = 50