

CS 211	Computers and Programming	Handout n
	Spring 2002	
	Prelim I	March 7, 2002

NAME:_____

CU ID:_____

Recitation instructor_____

Did you take CS 100M or 100J? If so, when?_____

You have one and a half hours to do this exam.

All programs in this exam must be written in Java.

Problem	Score
1	
2	
3	
4	
Total	


```

interface I1{
    int m1();}
interface I2 extends I1{
    int m2(); }
class B implements I1 {
    int i = 7;
    public int m1() {
        return i;}}
class C extends B implements I2{
    int i = -7;
    public int m2() {
        return i;}
    public int m1() {
        return i;}}
class Class1 {
    public static void main(String[] a) {
        B bee = new C();
        System.out.println(bee.i);    //will print -----
        System.out.println(bee.m1()); //will print -----
        System.out.println(bee.m2()); //will print -----

        I1 eye = (I1) bee;
        System.out.println(eye.i);    //will print -----
        System.out.println(eye.m1()); //will print -----
        System.out.println(eye.m2()); //will print -----

        C cee = (C) bee;
        System.out.println(cee.i);    //will print -----
        System.out.println(cee.m1()); //will print -----
        System.out.println(cee.m2()); //will print -----

        I2 me2 = (I2) bee;
        System.out.println(me2.i);    //will print -----
        System.out.println(me2.m1()); //will print -----
        System.out.println(me2.m2()); //will print -----
    }
}

```

2. (25 points) Use induction to prove the following fact. State carefully what the base and inductive cases are.

(a) (10 points) (easy)

$$\frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} + \dots + \frac{1}{(n-1)n} = \frac{(n-1)}{n}$$

(b) (15 points) (tricky)

$$\frac{1*3*5*\dots*(2n-1)}{2*4*6*\dots*2n} \leq \frac{1}{\sqrt{(2n+1)}}$$

This page left intentionally blank.

3. (25 points)

The following grammar describes arithmetic expressions that are similar to the ones discussed in lectures, except that function calls can appear as part of an expression. For simplicity, we will assume that a function can take only one parameter.

$$\begin{aligned} E &\rightarrow \textit{integer} \\ E &\rightarrow \textit{variable} \\ E &\rightarrow \textit{variable}(E) \\ E &\rightarrow (E + E) \end{aligned}$$

Here are some expressions that are legal in this language.

$(4 + (x + f(7)))$

$f((4 + y))$

Write a recursive program that takes a file as input, and returns true if the file contains a single expression that is legal according to this grammar, and false otherwise. You must use the CS211In class for reading from the file.

Here is the interface to the CS211In class.

```
interface CS211InInterface {
    int
        INTEGER = -1,
        WORD = -2,
        OPERATOR = -3,
        EOF = -4;
    int peekAtKind(); //returns one of the integers above
    int getInt(); //read an integer from the file
    String getWord();
    char getOp();
    boolean check(char c); //is the next token in the file operator c?
    boolean check(String s); //is the next token in the file word s?
    void pushBack(); //back up the file-pointer by one token in the file
    void close();
}
```

Here is a shell of the program to get you started.

```
static boolean parse(CS211In f) {
    boolean gotIt = getExp(f);
    if (f.peekAtKind() == f.EOF)
        return gotIt;
    else return false;
}
```

```
static boolean getExp(CS211In f) {
    switch(f.peekAtKind()) {
```

```
    }
}
```

This page intentionally left blank.

4. (20 points)

- (14 points) Write a public static Java method to convert a string containing only the characters 0 through 9 into its integer value. For example, given the string "0076", the method should return the integer 76. You will be graded for efficiency, so make your code as efficient as possible. You may find the following instance methods of the String class useful:
 - `int length()`: returns the number of characters in the string
 - `char charAt(int n)`: returns the character at position `n` of the string
- (6 points) Suppose `s1` and `s2` are two variables of type String. What do each of these constructs do?
 - `s1 = s2;`
 - `(s1 == s2)`
 - `(s1.equals(s2))`

This page intentionally left blank.