Cornell net id ____________________ **Name** ________________________________

CS 100J Prelim 2 17 March 2005

0	___________	out of 02
1	___________	out of 10
2	___________	out of 20
3	___________	out of 21
4	___________	out of 24
5	___________	out of 23
Total	________	out of 100

This 90-minute exam has 6 questions (numbered 0..5) worth a total of 100 points. We suggest that you spend a few minutes looking at all questions before beginning so that you can see what is expected. Budget your time wisely. Use the back of the pages, if you need more space.

Question 0 (2 points). Write your netid and your name, legibly, at the top of each page (Hint: do it now).

Question 1 (10 points). Answer the following questions briefly

(a) What is a loop invariant?

(b) Fill in the assignment so that the following assertion is true.

// {x is sum of 1..k}

x= ;

// {x is sum of 1..k+2}

(c) Write a **boolean** expression that evaluates to **true** when variable **x** is an instance of class **Car**, **false** otherwise.

Cornell net id ____________________ **Name** ________________________________

Question 2 (20 points). Write a complete class **MySum** that contains a static function **sumInteger** that returns the sum of the Integers in its parameter Vector **v**. You can assume that v is not **null**. **MySum** should contain only one thing: the static method.

Please write a **complete class** and **comments** (javadoc and coding comments).

Elements of the Vector can be of any class, however, function **sumInteger** should sum up only the Integers in the Vector and return the sum as an **int**. Examples: Below, we use ? for the value of an Object that is not of class Integer. For an object that is of class Integer, we use the integer.

If Vector v contains `(?, 8, ?, 1, ?, 4, ?)`, the output is 13.
If Vector v is empty, which we can write as `()`, the output is 0.
If Vector v is `(6, -1)`, the output is 5.

Here is a list of methods in class Vector that might be useful (not all of them are):

Return	**Method**	**Purpose**
Object	v.get(**int** k)	= v[k]
void	v.set(**int** k, Object ob)	replace v[k] by ob
int	v.size()	= the number of elements in v's list
int	v.indexOf(Object ob)	= i, where v[i] is ob

Here is a list of methods in class Integer that might be useful:

int	ob.intValue()	= the value of this Integer, ob, as an **int**

Question 3 (21 points). (a) What is the apparent type of a variable? What is the real type of a variable?

(b) Below are definitions of four classes.

```
public class Document {
   private String title;      // title of document
   private int numPages; // number of pages of
                                  // the document
  /** Constructor: document with title t and
        p pages */
   public Document(String t, int p) {
            title= t;
            numPages= p;
   }

  /** =  title of this document */
   public String getTitle()
       { return title; }

    /** =  number of pages of this document */
   public int getNumPages()
       { return numPages; }

   /** = the priority of the document */
   public int priority()
       { return 50 - numPages; }
}



public class Abstract extends Document {
   /** Constructor: an abstract with title t */
   public Abstract(String t)
       { super(t, 1);}
}
```

```
public class Letter extends Document {
   private String addressee;   // addressee of letter
   /** Constructor: letter with title t,  p pages,
         and addressee a */
    public Letter(String t, int p, String a) {
             super(t, p);
             addressee= a;
    }

    /** = "Letter" */
    public String letterType()
       { return "Letter"; }

    /** = the priority of the letter */
    public int priority()
       { return 100; }
}



public class Email extends Letter {
    /** Constructor: email with title t, p pages,
         and addressee a */
    public Email(String t, int p, String a)
       { super(t, p, a); }

    /** = "Email" */
    public String letterType()
       { return "Email"; }
}
```

For each of these pairs of statements, write down the value of d after execution. If the statements lead to an error, write "BAD" and briefly explain the error.

(1) Letter e= **new** Email("Prelim", 1, "gries@cs");
boolean d= "Email".equals(e.letterType());

Cornell net id ____________________ Name ________________________________

(2) Document c= **new** Letter(null, 3, "Gries");
int d= c.priority();

(3) Email b= (Email)(**new** Letter("", 2, "TAs"));
int d= b.letterType().length();

(4) Document a= **new** Abstract("Term Paper");
int d= a.priority();

(5) Document f= **new** Email("Reminder", 1, "me");
char d= f.letterType().charAt(1);

Question 4 (24 points). (a) Write an instance method equals(Object obj) for class Letter in Question 3. Here is the class definition, with only method equals for you to fill in.

```
public class Letter extends Document {
    /** = "obj is a non-null Letter with the same values in its fields as this Letter */
    public boolean equals(Object obj) {









      }
}
```

(b) Assume letter1 contains (the name of) a Letter object, draw the frame for method call letter1.equals(**null**).

Cornell net id ____________________ **Name** ________________________________

Question 5 (23 points). Write a method getTitleAcronym() in class Document in Question 3.
Field title in Documents contains only words and spaces that separate words.
A word is a sequence of non-space characters.
Between any two adjacent words there is exactly single space (the blank character).
There are no leading or trailing spaces in title.
Method getTitleAcronym() returns a string formed by the first character of each word in title.
Examples:
If title is “`Term Paper`”, the output is “`TP`”.
If title is “`trip to Florida during the spring break`”, the output is “`ttFdtsb`”.
If title is “`100 + 100 = 200`”, the output is “`1+1=2`”.

You will need to write a loop. A loop invariant is given below. You must use the given invariant to develop you loop. Your grades will be determined by how well you deal with the four loopy questions. Remember for a String object s, s.length() yields the length of s and s.charAt(**int** k) yields the character at position k.

```
public class Document {
      String title;  // title of the document
      /** = a String formed by the first character of each word in title.
            Precondition: title != null */
      public String getTitleAcronym() {
            //precondition: title != null



            //invariant: s contains the first characters of words in title[0..k-1]

            while (                                        )  {








            }
            //postcondition: s contains the first characters of words in title[0..]



      }
}
```