



ITI 1220/1620 Fall 2005
Introduction to Computing I

FINAL EXAMINATION

Length of Examination: 3 hours
Professor: Alan Williams

December 16, 2005, 09:30
Page 1 of 14

Identification: (complete this section using ink)

Family name : _____

Given names : _____

Student number : _____ Signature : _____

Instructions: Please read carefully!

1. This is a closed-book test. **No books, papers, calculators or other electronic devices are permitted.**
2. Answer all questions on the question sheet in the area provided. Les réponses en français sont acceptées. Questions answered in pencil will not be re-graded even if there is a marking error.
3. Algorithms are to be described using the format from the lectures and the notes.
4. You can use the back of the question sheet pages, or page 13, for calculations and other work. Pages 13 and 14 can be detached as they will not be marked.
5. If a question appears to be unclear, state a hypothesis and proceed to answer the question.
6. No student will be allowed to leave the exam room during the last 15 minutes of the exam.
7. When time is called at the end of the exam, **turn your paper over immediately. Stay in your place, and remain quiet until the proctors have collected all papers** and give clearance to leave.

1 (12)	2 (8)	3 (15)	4 (15)	5 (15)	6 (25)	7 (10)	Tot. (100)

Question 1A) (4 marks)

In this question, use only the following (**algorithm format**) Boolean expressions:

- comparison operators: <, >, =, ≤, ≥, and ≠
- Boolean logical operators: NOT, AND, OR
- arithmetic operators: +, -, *, /, and MOD (modulo)
- variable names and constants.

Use parentheses where necessary. Do **not** use Java syntax!

The Ontario “Drive Clean” program requires that a car must pass an emissions test for licence renewal if both of the following conditions are met:

- The car’s model year MY is odd and the current year CY is even, or vice versa
- The difference between the current year and the model year is at least 3 but no more than 20.

Write a Boolean expression that is true if a car needs a Drive Clean emissions test, and false otherwise.

Answer:

Question 1B) (4 marks)

What will be printed by the following program? Write your answer below the program.

```
class ABC
{
    public static void main(String[ ] args)
    {
        char[][] x = {{'i', 'j'}, {'4', '5'}};
        char[][] y = {{'x', 'y'}, {'a', 'b'}};
        int i;
        char t;

        for (i = 0; i < 2; i=i+1)
        {
            t = y[(i+1)%2][1];
            y[i][(i+1)%2] = x[i][0];
            y[(i+1)%2][1] = t;
        }

        System.out.print(y[0][1]);
        System.out.println(y[1][0]);
    }
}
```

Write your answer here:

Question 1C) (4 marks)

Consider the following two classes:

```
class C1
{
    private int[] v1 = {1,4,9};
    public int v3;

    public static int m1 (C2 p)
    {
        ...
    }

    private C1 m2 (int m)
    {
        ...
    }
}

class C2
{
    public static char v3;

    public C2 (int n)
    {
        ...
    }

    private void m3 (int m)
    {
        ...
    }
}
```

Suppose that the following instructions are used in the `main()` method in a class `Test`. Each choice should be considered independently – as if it were in its own `main()` method. Circle the letter of the statement which does NOT cause an error.

- (a) `int v2 = C1.m1(this);`
- (b) `char v3 = C2.v3;`
- (c) `C1 x = new m2(4);`
- (d) `C1 w = new C1();`
`w.v1[2] = 3;`
- (e) `C2 y = new C2();`
`int z = C1.m1(y);`

Question 2: (8 marks)

Here is a program that uses recursion.

```
class AClass
{
    public static void main(String[ ] args)
    {
        aMethod(137210);
        System.out.println( );
    }
    public static void aMethod(int i)
    {
        if (i == 0)
        {
            ; // do nothing
        }
        else
        {
            if (i%10 == 1 )
            {
                System.out.print( "one " );
            }
            else
            {
                if ( i%10 == 2 )
                {
                    System.out.print( "two " );
                }
                else
                {
                    if ( i%10 == 3 )
                    {
                        System.out.print( "three " );
                    }
                    else
                    {
                        ; // do nothing
                    }
                }
            }
        }
        aMethod(i/10);
    }
}
```

What is printed during the execution of the method main in this class?

Answer :

Question 3A): (10 marks)

Translate the algorithm on page 13 to a **Java method**.

Question 3B) (5 marks)

Design a starter method that calls the method in question 3A). The starter method should be able to be called as follows: `isPalindrome = palindrome(aString)`

Question 4: (15 marks)

In the upcoming election, the candidate who has the most votes in an electoral district wins the riding and becomes the Member of Parliament. However, if the difference in the number of votes between the top two candidates (that is, the candidates with the largest and second-largest numbers of votes) is very close, a recount of the votes will take place.

Design an algorithm that will take an array **Votes** containing the number of votes for each of the **N** candidates in an electoral district, and a value **Diff** where if the difference in the number of votes for the top two candidates is less than or equal to this value, a recount will take place. The algorithm should return true if the votes should be recounted and false otherwise. The array **Votes** is **not** sorted.

[Use this page and the next for the answer; it is suggested to put the body on the next page.]

Question 4 continued:

Question 5: (15 marks)

A "magic square" is a square matrix of integers where every row and every column can be summed to the same single value. For example, in the matrix A below, the sum of every row and column is 15.

$$A = \begin{bmatrix} 2 & 9 & 4 \\ 7 & 5 & 3 \\ 6 & 1 & 8 \end{bmatrix}$$

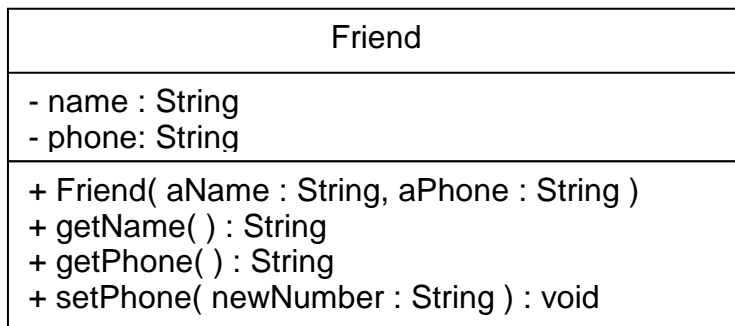
Write a Java method that will take a square matrix of integers A and returns true if A represents a "magic square" and false otherwise. Your method should be efficient.

```
public static boolean isMagic( int[][] a )
```

Question 6: (25 marks)

In this question, you will create a class `AddressBook` that will be able to store a collection of `Friend` objects.

The class `Friend`, which has already been implemented, provides storage for a person's name and phone number. The class contains a constructor, and two accessor methods. A UML class diagram for the class `Friend` is as follows:



On the next two pages, you will fill in the methods for `AddressBook`. Your `AddressBook` class should provide a constructor and three public methods that would permit the following class `TestAddressBook` to execute:

```
class TestAddressBook
{
    public static void main (String[] args)
    {
        AddressBook myBook = new AddressBook( 2 );
        myBook.addFriend( new Friend( "Alice", "555-1212" ) );
        myBook.addFriend( new Friend( "Tommy", "555-3434 " ) );
        myBook.addFriend( new Friend( "Pizza", "737-1111" ) );

        myBook.changePhone( "Tommy", "867-5309" );
        myBook.print();
        myBook.changePhone( "Pizza", "310-1010" );
    }
}
```

Executing `main()` would result in the following being printed on the screen :

```
The address book is full.
Name: Alice, Phone: 555-1212
Name: Tommy, Phone: 867-5309
Pizza is not a name in the address book.
```

Question 6:(continued)

Complete the class `AddressBook` on this page and the next.

```
class AddressBook
{
    // ATTRIBUTES: (3 marks)

    // CONSTRUCTOR: (5 marks)
    // Takes one integer parameter representing the maximum number of entries that can be put into
    // the address book.

    // METHOD addFriend: (6 marks)
    // Method parameters: a Friend object that should be added to the AddressBook.
    // Results: will print a message if the address book is full (see page 9 for message format)
    // Modified: the AddressBook object
```

Question 6:(continued)

```
// METHOD print: (5 marks)  
// Method parameters: (none)  
// Returns: (none)  
// This method prints a list of names and phone numbers in the address book.  
// See the TestAddressBook sample output on page 9 for format.
```

```
// METHOD changePhone (6 marks)  
// Updates the phone number of the first Friend with the specified name.  
// Method parameters: a String which is the name of a Friend, and a String which is a  
// new phone number for that Friend.  
// Result: an error message if the friend is not in the address book (see page 9 for format)  
// NOTE: recall that two strings s1 and s2 can be checked for equality with s1.equals(s2)
```

```
} // End of class AddressBook
```

Question 7: (10 marks)

Write a **recursive** method that takes as a parameter a non-negative integer and generates the following pattern of stars. You may use a loop to generate one line of stars, but **not** the entire pattern. If the non-negative integer is 4, then the pattern generated is:

```
* * * *
* * *
* *
*
*
* *
* * *
* * * *
```

The header of the method should be the following:

```
public static void stars(int n)
```

This page is for calculations and other work, and can be detached. This page will not be marked.

Algorithm for Question 3 (this page can be detached)

This algorithm determines whether a given phrase is a **palindrome** – a phrase that reads the same when the characters in the phrase are reversed. For example, “step on no pets” is a palindrome.

GIVENS:

Phrase: (an array of characters representing a phrase to be checked)
 Left: (index of leftmost character to check in Phrase)
 Right: (index of rightmost character to check in Phrase)

RESULT:

IsPal (true if Phrase is a palindrome, and false otherwise)

INTERMEDIATES:

Length (length of the interval examined in Phrase)

HEADER: IsPal \leftarrow PalindromeRec(Phrase, Left, Right)

BODY:

