

# ITI 1120 Fall 2012

## Introduction to Computing I

### Midterm Examination - Solution

**Length of Examination: 1.5 hours**

**Nov 3, 2012, 10:00**

**Professors: Gilbert Arbez, Morvarid Sehatkar**

**Page 1 of 12**

**Instructions: Please read carefully!**

- 1. Complete all sections of the identification area in ink.**
- 2. This is a closed-book test. No books, papers, calculators or other electronic devices are permitted.**
- 3. Answer all questions on the question sheet in the area provided. Questions answered in pencil will not be re-graded even if there is a marking error.**
4. The marks allocated to each question are indicated. Not all questions are worth the same amount, so plan your time accordingly. The midterm will be scored out of 40 marks, which represents 20% of your final grade.
5. Algorithms are to be described using the format from the lectures and the notes.
6. You can use the back of the Exam Annex pages for calculations and other work.
7. Les réponses en français sont acceptées.

**Identification:**

Name: \_\_\_\_\_

Student number: \_\_\_\_\_

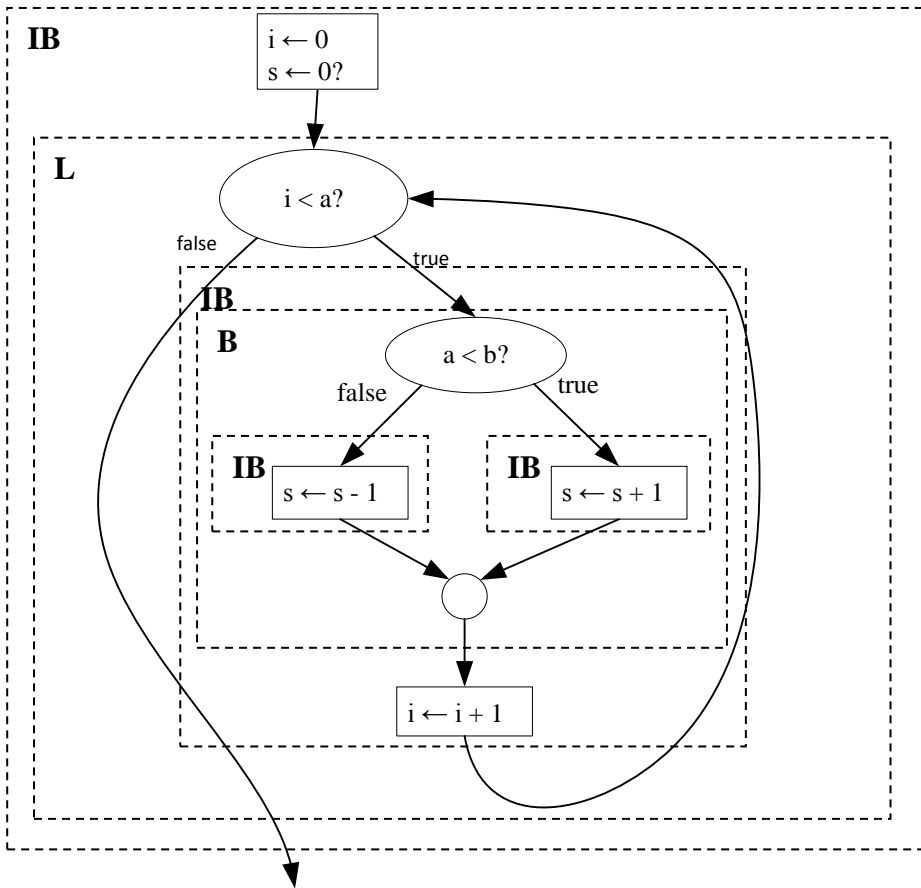
**For use of grader:**

Question	Marks available	Marks received
1	8	
2	8	
3	12	
4	12	
Total	40	



**Part b)**

i) Circle all the instruction blocks in the algorithm below and label them with the letter IB. Circle all the branch control instructions and label with B. Circle all the loop control instructions and label with L.



**0.5 point for each correctly identified instruction block for a total of 3 points**

```

i = 0;
s = 0;
while(i < a)
{
    if(a < b)
    {
        s=s-1;
    }
    else
    {
        s=s+1;
    }
}
    
```

ii) Briefly describe why the translation of the algorithm body diagram to Java is incorrect:

**The instruction blocks for the branch instruction are reversed, that is**

- **the false instruction block in the algorithm branch is translated to the true instruction block of the Java if statement**
- **the true instruction block in the algorithm branch is translated to the false instruction block of the Java if statement**

1 point

## Question 2 (8 points in total – 4 points for each part)

### Part a)

The value of  $\pi$  can be approximated by the following series:

$$\pi = 4 \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \dots \right) \quad (\text{Eq. 1})$$

$$= 4 \sum_{i=1}^N \left( \frac{1}{2i-1} - \frac{1}{2i+1} \right), \quad \text{for odd values of } i \quad (\text{Eq. 2})$$

The following problem solving Java method estimates  $\pi$  for a given value of n (represents N summation limit in Eq. 2). Find the three logic errors in the method and correct.

```

/*-----
 * Method: estimatePI
 * Description: Provides an estimation of the value for PI.
 * Parameters (givens):
 *     n - the limiting value for the summation of the terms.
 * Assumptions: n is odd.
 * Calculation based on the equation:
 *     pi = 4 * sum for i=1 to n (i odd) of the terms: 1/(2i-1) - 1/(2i+1)
 */

```

```
public static double estimatePI(int n)
```

```

{
    // Declaration of variables
    double pi;           // RESULT: for computing
    int i;               // the variable i in the
    double tminus, tplus; // the terms 1/(2i-1)
    double diff;         // difference tminus-tplus
    // Body
    pi = 0.0;
    i = 1;
    // Loop for summing terms
    while(i<=n)
    {
        // Add the term at value i
        tminus = 1.0/(2.0*i - 1.0);
        tplus = 1.0/(2.0*i + 1.0);
        diff = tminus - tplus;
        pi = pi + diff;
        i = i + 2;
    }
    pi = 4.0*pi;
    // Return the results
    return(pi);
}

```

Should be <= so that the value of n is included in the calculation. (1 point)

Need instruction  $i = i + 2;$  to increment i at each iteration of the loop by 2 (2 points)  
Give 1 mark if answer  $i = i + 1;$

Need instruction  $pi = 4*pi;$  to complete Eq. 2. Also valid to update instruction in loop instruction block:  $diff = 4*(tminus-tplus);$  (1 point)

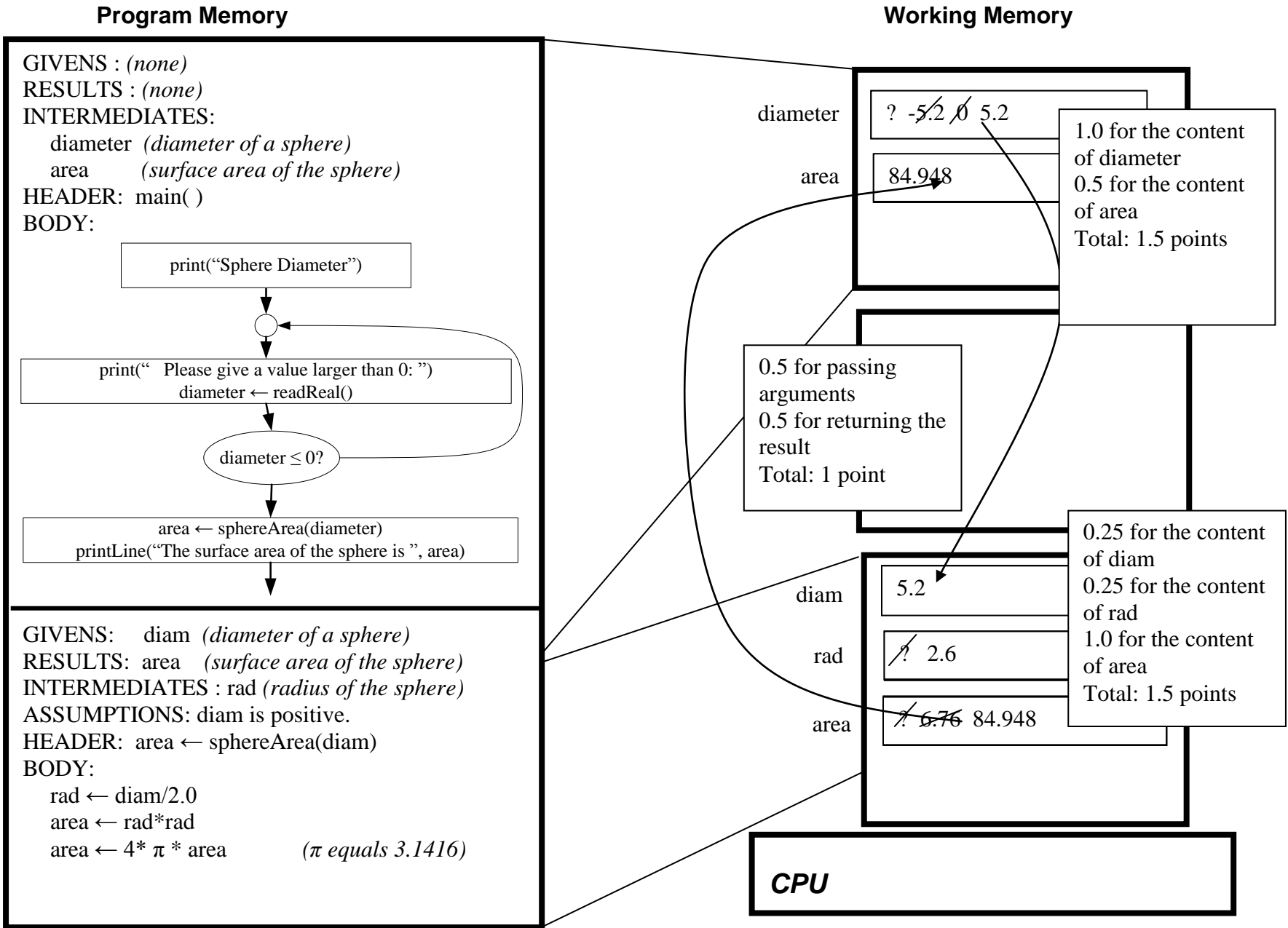
**Part b)**

The following example output is produced by the execution of the program given on the next page.

```
Sphere Diameter
Please give a value larger than 0: -5.2
Please give a value larger than 0: 0
Please give a value larger than 0: 5.2
The surface area of the sphere is 84.9486
```

On the following page, show how the contents of the working memory are changed by the execution of the algorithm model as follows: (1) show how values are passed from one algorithm model to the other using arrows; (2) record successive assignments to variables/parameters as follows:

**Variable** *d, r, s, A, 10*



Some calculations:  $5.2/2 = 2.6$      $2.6^2 = 6.76$      $4 \times \pi \times 6.76 = 84.9486$

**Question 3 (12 marks)**

Translate the algorithms found in the exam annex to a Java program by completing the partially filled `main( )` and `translateGrades( )` methods below and on the next page. You do not need to add an identification section to the program.

```

public class Q3
{
    /**
     * Read in a set of numerical grades and translate to letter grades
     */
    public static void main(String[] args)
    {
        // DECLARE VARIABLES / DATA DICTIONARY
        double [] numericalGrades; // For referencing array of numerical grades
        char [] letterGrades;      // For referencing array of letter grades
        int ix;                    // For indexing into arrays
        int n = numericalGrades.length; // Length of the array

        // Get letter grades from user
        System.out.println("Please give a set of numerical grades");
        numericalGrades = ITI1120.readDoubleLine();
        // Translate to letter grades
        letterGrades = translateGrades(numericalGrades);
        // Display to user
        System.out.println("Translated Grades");
        for(ix=0; ix < n; ix++)
        {
            System.out.println("    " + numericalGrades[ix] + "->"
                               + letterGrades[ix]);
        }
    }
}

```

Can also use `numericalGrades.length` instead of declaring `n`

Input :  
1 point  
Can use `Scanner` instead of `ITI1120`

Declarations :  
1 point

Call to method :  
1 point

Loop : Can also use a while loop.  
1 point

Output:  
1 point

```

/*-----
 * Method: translateGrades
 * Parameters (Givens):
 *     numGrades - reference to numerical grades
 *               - numGrades.length gives the length of the array
 */
public static char [] translateGrades(double[] numGrades)
{
    // Declaration of variables
    char [] letGrades; // RESULT: reference to array of letter grades
    int index; // For traversing the arrays
    int n = numGrades.length; // number of elements in the arrays
    // Body
    letGrades = new char[n];
    index = 0;
    while(index < n)
    {
        if(numGrades[index] > 85.0)
        {
            letGrades[index] = 'A';
        }
        else
        {
            if(numGrades[index] >= 70.0)
            {
                letGrades[index] = 'B';
            }
            else
            {
                if(numGrades[index] >=50.0)
                {
                    letGrades[index] = 'C';
                }
                else
                {
                    letGrades[index] = 'F';
                }
            }
        }
        index = index + 1;
    }
    return (letGrades);
}
}

```

Header :  
1 point

Declarations :  
1 point

Can also use numGrades.length instead of declaring n

Loop : Can also use a for loop  
1 point

Branch :  
1 point

Branch :  
1 point

Branch :  
1 point

Return results :  
1 point

**Question 4 (12 marks)**

The weight of a child on has been recorded for each birthday, starting from age 0 (the day of birth) up to their current age; these values are stored in an array referenced by the variable `weight` (thus the indexes of the array represents the child's age).

Complete the algorithm, *greatestIncrease*, to determine at which age the child had the greatest percentage increase in weight from one birthday to the next. For example, if the greatest increase was between the 13<sup>th</sup> and 14<sup>th</sup> birthdays, the result of the algorithm should be 14.

The greatest percentage increase is calculated relative to the first birthday. For example, if the weight of the child at his/her 13<sup>th</sup> birthday is 40 kg and at his/her 14<sup>th</sup> birthday is 45 kg, then the percentage increase is  $(45-40)/40 = 0.125$ , or 12.5%.

**Use only the algorithm syntax presented in class. Any use of Java in this question shall lead to a ZERO for the question.**

GIVENS:

`weight`            (reference to array of weight values)  
`n`                    (number of elements in the weight array )

RESULTS:

`ageGreatestInc` (age at which the maximum increase occurred)

INTERMEDIATES:

`ix`                    (**Index into the array**)  
`greatestInc`        (**for determining the largest increase**)  
`inc`                    (**for computing the increases.**)

*Intermediates :*  
*2 point*

HEADER: `ageMaxIncrease` ← `findAgeWithLargestIncrease(weight, n)`

BODY:

*Header:*  
*1 point*

