

Section, Name and ID#:

AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
CS 120 Introduction to Object-Oriented Programming
MIDTERM EXAM

Date / Time:

Friday, March 17 2017 at 17:30

Duration:

2 hours

Attention:

ANY TYPE OF COMMUNICATION IS STRICTLY PROHIBITED

Write down your section, name and ID# at the top of all used pages

Participation:

Problem 1: Consider below a C++ function `float kahan(float num1, float num2, float& compensation)` that implements the *Kahan Summation Algorithm* for high-precision compensated summation of two float arguments `float num1` and `float num2`:

```
float kahan(float num1, float num2, float &compensation)
{
    float result;
    num2 -= compensation;
    result = num1 + num2;
    compensation = (result - num1) - num2;
    return result;
}
```

Using this function, write a C++ function `float e(int n)` that computes the value e by the following formula:

$$e = \sum_{k=0}^n \frac{1}{k!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1*2} + \frac{1}{1*2*3} + \dots$$

Recall that the factorial of non-positive numbers equals to 1 by definition.

The initial value of `float compensation` is 0.0.

```
float e(int n) {
    float form ans = 0;
    for (int i = 1; i < n+1; i++) {
        int number = 1;
        for (int j = 1; j < i; j++) {
            number = number * j;
        }
    }
}
```

Use the backside, if needed

use kahan theorem. here
at the end.

Problem 1 of 4

00P.MT.170317.1044

Section, Name and ID#:

Problem 2: Write a Java method `public static double[] mean(double[] data)` that takes as its argument an array of data points `double[] data`, and returns a two-element array – the first element being the mean value of the data points and the second element being the standard deviation. The standard deviation σ of n numbers a_i is computed as:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{n-1} (a_i - \text{mean})^2}{n}}$$

```
public static double[] mean(double[] data) {
```

```
    double mean = 0;
```

```
    double s = 0;
```

```
    int length = data.length;
```

```
    for (int j = 0; j < lengthlength; j++)
```

```
        s = data[j];
```

```
}
```

```
    mean = s / length;
```

```
    double t = 0;
```

```
    double a = 0;
```

```
    for (int j = 0; j < length; j++)
```

```
        float a = data[j] - mean;
```

0
see TT