## 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 pi(int n) that computes the value  $\pi$  by the following formula:

$$\pi = 2\sum_{k=0}^{n} \frac{(2k-1)!!}{(2k)!!(2k+1)} = \frac{2}{1*1} + \frac{1}{2}*\frac{2}{3} + \frac{1*3}{2*4}*\frac{2}{5} + \frac{1*3*5}{2*4*6}*\frac{2}{7} + \cdots$$

Recall that n!! is the product of odd numbers from I to n, if n is odd; and is the product of even numbers from 2 to n, if n is even. The double factorial of non-positive numbers equals to I by definition.

The initial value of float compensation is 0.0.

int Pouble fuet (int n) {
if (n % 2 - 1 5) for ( int i = 1 : i < n ; i or ) d forlint i=2; i in; iri)

n-fuet\* = i; j

We if (n = 0) d n-fuet = 16 return n. fuet: %

float pilinta) of
float presult 1; result 2; result 3; result 3; result 4; result 4; result 3; result 3; result 3; result 5; result 6; r result 2 = result3; } result: kukun (result1, result2, D.D): 3 return result; 4

Use the backside, if needed

Problem 1 of 4

OOP. MT. 170317, MO70

Problem 2: Write a Java method public static double[] lin(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 slope of the linear regression and the second element being the intercept. The linear regression approximates the data points by the linear formula

y = kx + b

where the slope k and the intercept b are computed as

 $k = \frac{\overline{xy} - \overline{x} \ \overline{y}}{\overline{x^2} - \overline{x}^2}, b = \overline{y} - k \ \overline{x}$ 

Here  $\bar{x}$  is the mean of the x coordinates,  $\bar{y}$  is the mean of the y coordinates,  $\bar{x}^2$  is the mean of the squares of the x coordinates, and  $\bar{x}y$  is the mean of the products of the x and y coordinates. Use the element indices of the array double[] data as x coordinates and the element values as y coordinates. You may assume and use the method double mean(double[] a).

public static double I lin I double I duta) } doubles result - new double [2]: ant six = new x[duta length]; intsix'd = new Entsduta length]; double Is = new ys duta length;

soubly my = new xy l duta length;

for (int i=0; i < duta length; re)

for (int i=0; i < duta length; x-2[i] = i# ; 1 Uli] = duta [i]: For (dnt j. 0; j < duta. Bength; jee) & for lint 2k = 0; k = duta length; kee ) d resul[0] = (mean (xy) - (mean(x) # mean (y))) / (mean (x-2) -- (meun (x). meun (x))); resul[]] = mean(y) - (result[0] \* mun(x)); return result; ( Problem 2 of 4 Use the backside, if needed ADP.MI. 120317 MOJO

Problem 3: Write a Java function *public static double area(double]]]] vertex)* that takes as its argument a 2-by-n array of a convex polygon's vertex coordinates *double*]]] vertex – the x coordinates in the first row and y coordinates in the second-row. It returns polygon's area as follows:

- 1. Divides the polygon into triangles by connecting the first vertex with the  $n^{th}$  and  $(n+1)^{st}$  vertices;
- 2. Adds the areas of the constructed triangles using the formula  $area = \sqrt{p(p-a)(p-b)(p-c)}$ , where a, b and c are the sides and p = (a+b+c)/2.

You may assume and use a method double dist(double x1, double y1, double x2, double y2) that takes as its arguments coordinates of two points and returns the distance between them.

public stufic obuble used (double [ ]] perlog

coable areaco) a, b, c) p;

for (int i: 0; ic vertex. length 1 is o) {

u=dist (vertex [ O [ 0] vertex [ 1] [ 0] ) vertex [ 2] [ 1] );

vertex [ 1 ] [ i ] ]; b=dist (vertex [ 0 ] [ i] , vertex [ 1] [ i ] );

vertex [ 1 ] [ i o ] ] , vertex [ 1 ] [ i o ] );

c= dist (vertex [ 0 ] [ 0 ] , vertex [ 1 ] [ i o ] );

vertex [ 1 ] [ i o ] ];

vertex [ 1 ] [ i o ] ];

p=(u+6 o e) / 2;

aveue=syllot (p-a) (p-6) (p-c)); g

return area;

Problem 4: Write a Java method public static void magic4N(int[][] square) that creates a magic square of a 4N-by-4N size using the following algorithm:

1. Creates an array of the same size as int[][] square and fills it forward with successive integers assigning I to the top-left element;

2. Creates anther array of the same size as *int[][] square* and fills it backward with successive integers assigning I to the bottom-right element;

3. Divides the original int[][] square into 16 blocks of the same size – 4 blocks per row and column. In the on-diagonal (shaded) blocks copies the elements from the first array, and in the off-diagonal

blocks	diagonal (sha	, ,	ks copi	es the e	ements	rom t	he fir	st arra	ay, an	d in t	the of	f-diag	onal	
00	copies the elei	ments from	n secon	d array.		-1		4						
9	0 10 11	91 5	6		D-600			/62	61	60	59			
-		20 101		15 1				54	53	52	51)			
Jun 1	19 19 15 16 7 27	20 21 29	TARREST PLANS AND ADDRESS OF THE PARTY NAMED IN		6	48	47	74	× 1.	504		42	41	11
27	3 34 35	28 29 36 37	The second second	3/ 3		40	39					34	33	60
1.8	43	44 45	THE RESERVE OF THE PARTY OF THE	39 6	- Comment	32	23			4.1		18	25	
4		5 2 3	-	55 5	- Control	24	2.5	14	13	12	11/	10		
5	7 58 53	00 5	161	William Company of the Company	4			6	5	4	3			
0	. Min d				1									0
<i>P</i>	ublic st at arr s at urr 2 r (int i	unc	100	1 m	ugies	11/	int	II	[]	syu	ure	11		
ir	if arr s	1 - neu	rin	48 squ	1050 B	PI IS T	(71.		rt.	1	- 100		Nº.	
C	urr_2	= neu	Fin	16 89	uure	De se	1 4 628	Mure	101.6	conth	7:	W ()	n 71 8	Burgas
101	t urr_2 r lint i	10,	csai	auro	Po	CERT S	MA.	168	War	toy.	lenal	147.	durp!	01 (and)
		1 - 1	1 1			1-2	) [ 1	( )	1		6			
	urr.	15:1	17.	sg ucu	reli	·le	ngti	4:0	00	14				
	4	J[i]	177 =	511;	9									
Soi	clint in	n												
F	rlint is	110	zg Wu	11.6	could	, , ,	, )		1	DP.	mT.	170	03/4	moso
														1-704
			-0	mal 11	1 1 0	11								
	urr-25-	11.1-			0.00	agin	150	11).	1					
	urr-25i	11/1:	Sgua	resil	· ann	ngin U	1,50	(1).	1					
	5		- 9 - N	1121	· Clagi	7 -	1.1	; 9	1					
	forlint	C=1) . ;	-9-4	1	· ling	4	1-1	; j	1					
	for lint for lint	i=0; i	Car	. len	· Bengi	4	1-1	; j	1	1	10	11		>
	for lint for lint	i=0; i	Car	. len	· Bengi	4	1-1	; j	1	4	el:	Jug	re-	>
	for lind for lind arri	i=0; i t j=0;	car	. len	· Bengo	(A Pt) Pt)	1-1	; 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lint for lint arri	i=0; i t j=0; illj=ar	( ar	. len	· Cengi	(A	1-1	3 9		4				
	for lind  for lind  urrl  urrl  errl  errl  length ruge.  lint j = 0  list j = 0	ististarios de la constanta de	correction of the service of the ser	length length wir gth pu	gth/4)  gth/4)  gth/4)  length  length  length	1 - ( e e ) ( j ) ( ) ( ) ( )	1-1 2 ur 143 Bery 151 151	r. j urr gth ingth	lar-di- un.	Ilun Ilun Iil ell Vir Venyt Pr				
	for lind  for lind  urrl  urrl  errl  errl  length ruge.  lint j = 0  list j = 0	ististarios de la constanta de	correction of the service of the ser	length length wir gth pu	gth/4)  gth/4)  gth/4)  length  length  length	1 - ( e e ) ( j ) ( ) ( ) ( )	1-1 2 ur 143 Bery 151 151	r. j urr gth ingth	lar-di- un.	Ilun Ilun Iil ell Vir Venyt Pr				
	for lind  for lind  urrl  urrl  errl  errl  length ruge.  lint j = 0  list j = 0	ististarios de la constanta de	correction of the service of the ser	length length wir gth pu	gth/4)  gth/4)  gth/4)  length  length  length	1 - ( e e ) ( j ) ( ) ( ) ( )	1-1 2 ur 143 Bery 151 151	r. j urr gth ingth	lar-di- un.	Ilun Ilun Iil ell Vir Venyt Pr				
	for lind  for lind  urrl  urrl  errl  errl  length ruge.  lint j = 0  list j = 0	ististarios de la constanta de	correction of the service of the ser	length length wir gth pu	gth/4)  gth/4)  gth/4)  length  length  length	1 - ( e e ) ( j ) ( ) ( ) ( )	1-1 2 ur 143 Bery 151 151	r. j urr gth ingth	lar-di- un.	Ilun Ilun Iil ell Vir Venyt Pr				
	for lind  for lind  arri  uref  erri  length puge.  lint j = 0  rli ] [ ] ]  de, if needed  arri  arri	i=0; i t j=0; illj=ar	Carification of the art the ar	length length light eur	gth 14 - ur-le length 14 - ur-le length 14 - urr-length 14 urr-2	1	dery length	r. f. urr gth ingth ca.	lastingth -	I lun I lun	gth- ength:	1. ler 141 141 140f		