09/12/2020

CSE1141 Midterm Exam – Fall 2020 (Online Exam)
(Duration: 100 minutes)

PLEASE WRITE YOUR NAME, SURNAME, AND THE HONOR CODE BELOW IN YOUR HANDWRTING AND SIGN YOUR ANSWER SHEET.

HONOR CODE: "On my honor, I have neither given nor received any unauthorized and/or inappropriate assistance for this exam. The work done on this exam is totally my own. I understand that by the school code, violation of these principles will lead to a zero grade and is subject to harsh discipline issues."< your signature>

<u>Submission Instructions:</u>

- **You will write your answers on A4 size white blank sheets, with a legible handwriting.**
- **You will scan all your answer sheets using a (free) mobile phone scanner application into a single pdf file. You must use the PDF format.**
- **The name of the file should be "student name_surname.pdf".**
- **You have to upload your single pdf file to UES before the deadline.**
- **In case of last minute problems (in case of not being able to upload to UES), the answer sheets should be sent by email without exceeding the deadline. (If you cannot upload it to UES, you can upload it to Canvas without exceeding the deadline).**
- **The answer sheets sent after the deadline will not be accepted.**
- **The maximum number of uploads is limited to 2.**

Q1. (28 points) For each of the following java expressions, determine the **type** (e.g. int, boolean, String, double or char) and **value** of each of the expressions listed below. For example, if the expression were 6+3 the **type** would be **int** and the **value** would be **9**.

Just write Q1.x (x can be from 1 to 14) and your corresponding answers to the answer sheet. (2 pts. each)

	Expression	**Data type**	**Data value**
1.	`'2' <= 'A'`		
2.	`"3" + 5 + 0.16`		
3.	`Math.max(6,("engineering").charAt(3))`		
4.	`9 / 2`		
5.	// Suppose `ID` is a long with a value of `YOUR STUDENT ID`. `ID % 5`		
6.	`(3 * 2) / 3 > 8`		
7.	`Math.round(50.5)`		
8.	`Math.rint(50.5)`		
9.	`Math.ceil(10.1)`		
10.	`Math.floor(9.9)`		
11.	// Suppose `ID` is a long with a value of `YOUR STUDENT ID`. `((ID % 2 == 0) ? "even" : "odd")`		
12.	// Suppose `foo` is a string with a value of “`CSE2020`” `foo.indexOf("E");`		
13.	// Suppose `foo` is a string with a value of “`CSE2020`” `foo.length()`		
14.	// Suppose `s` is a string with a value of “`CSE1141`” `s.substring(`**`3`**`, `**`6`**`)`		

Q2. (28 points)

a) **(4 pts.)** If **a**, **b** and **c** are boolean variables, for what values of **a**, **b** and **c** does the expression

```
( (!a && b) || c )
```

evaluate to true? Please write all possible values! You can write “Don’t care” if you think that a variable can take any values (true or false) and the result will not change based on its value.

b) **(4 pts.)** Rewrite the following statement using if-else statements.

```
String s1, s2;
s2 = s1.length() == 0 ? "hello" : "goodbye";
```

c) **(4 pts.)** Consider the two code segments below (a and b), where **x** and **y** are **int** variables:

a)	b)

`if (x > y) {`
`  x = x / 3;`
`}`
`else {`
`  y = y + 2;`
`}`
	`if (x > y) {`
	`  x = x / 3;`
	`}`
	`if (x <= y) {`
	`  y = y + 2;`
	`}`

Will these two code segments *always* produce the same assignments for **x** and **y** in the end, or are there certain values of **x** and/or **y** where they will behave differently? If you believe that they will behave the same way, provide a convincing argument in 1-2 sentences to explain why you believe that this is so. If you believe that they will behave differently, provide at least one example of values for **x** and **y** where they behave differently and describe the difference.

d) **(4 pts.)** What is `y` after the following **`switch`** statement is executed? Rewrite the code using an **`if-else`** statement.

```
int x = 3, y = 3;
 switch (x + 3) {
   case 6: y = 1;
   default: y += 1;
 }
```

e) (12 pts.) Consider the following code segment where **x** is a variable of type **int**:

```
if (x > 45 ) {
  if (x < 60) {
    System.out.println(“Pound Cake”);
  }
  else {
    System.out.println(“Red Velvet Cake”);
  }
}
else if (x > 32) {
  if (x < 50) {
    System.out.println(“Carrot Cake”);
  }
  else {
    System.out.println(“Sponge Cake”);
  }
}
else if (x < 15) {
  System.out.println(“Yellow Butter Cake”);
}
else {
  System.out.println(“Flourless Cake”);
}
```

For each of the following output values, give the range of values for the variable **x** that will cause the segment to display the given String. For example, if anything between 0 and 10 including both 10 and zero will produce the string, your answer should be **0 <= x <= 10**. If a value can never be produced by this code, you can write “**Not possible!**”. Suppose that the smallest value of **x** can be **0**.

1) `"Pound Cake"`

2) `"Red Velvet Cake"`

3) `"Carrot Cake"`

4) `"Sponge Cake"`

5) `"Yellow Butter Cake"`

6) `"Flourless Cake"`

Q3. **(12 points)** Please analyze the code segments below and indicate if there are any possible problems with the code. Explain the cause of problems, or write "No problem" if you do not see any problems. In case of "No problem" please write the output of the program.

(Please note that the code segments in **a** and **b** are properly placed in a main function in a class).

a) (4 pts.)

```
double sum = 0;
double d = 0;
while (d != 10.0) {
  d += 0.1;
  sum += sum + d;
}
```

b) (4 pts.)

```
int i = 0;
for (i = 0; i < 10; i++);
  System.out.println(i + 4);
```

c) (4 pts.)

```
import java.util.Scanner;

public class Test {
  public static void main(String[] args) {
    int sum = 0;
    for (int i = 0; i < 100000; i++) {
      Scanner input = new Scanner(System.in);
      sum += input.nextInt();
    }
  }
}
```

Q4. (32 points)

a) **(8 pts.)** For the questions below, use the following variable declarations. As a reminder: `strictly` larger or smaller than x means that x is not included, and `strictly` between x and y means that the value is between x and y but does not include x and y:

```
int max, min;
boolean stop;
String bar;
```

- (3 pts) Write a `while` condition for a loop that runs as long as **max** is strictly greater than **min** and also less than the length of the string **bar**.

  while (____)

- (5 pts) Write a while condition for a loop that runs as long as **`stop`** is false, **`min`** is strictly less than 6, and the character at position 3 in the String **`bar`** is not a `'q'`.

  while (____)

b) **(9 pts)** Write a for loop that computes 1/3 + 2/4 + 3/5 + ... + 99/101 and store the result in a double variable `sum`.

```
double sum = 0;
for (                                        ) {




}
System.out.println("Sum is " + sum);
```

c) **(15 pts)** Suppose that the pattern below is printed by a program that uses nested loops.

			1			
		1	2	1		
	1	2	4	2	1	
1	2	4	8	4	2	1

Please fill in the blanks in code below to print the corresponding patter. (Each cell represents the position of a character in the pattern above).

```
int number = 0;
for (int row = 0; ______________; row++) {

    for (int col = 1; ______________ ; col++)
        System.out.print(_________________________);

    for (int col = 0; __________; col++) {

        number =_____________________________

        System.out.print(number);
    }
    for (int col = __________; col >= 0; __________) {
        number =_____________________________
        System.out.print(number);
    }
    System.out.println();
}
```

SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

`String Methods:`

– `charAt(int index):` Returns the character at position index in this String.
– `indexOf(char ch):` Returns the index position of the first occurrence of the character ch within this String. If the character does not occur in this String, the method returns -1.
– `indexOf(String str):` Returns the index position of the first occurrence of the String str within this String. If the String does not occur in this String, the method returns -1.
– `length():` Returns the length of this String.
– `substring(int beginIndex):` Returns a new string that is a substring of this String starting from index beginIndex.
– `substring(int beginIndex, int endIndex)`: Returns a new string that is a substring of this String starting from index beginIndex to index endIndex-1.

`Math Methods:`

- `random():` Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
- `pow(double a, double b):` Returns a raised to the power of b (a^b).
- `ceil(double x):` x rounded up to its nearest integer. This integer is returned as a double value.
- `floor(double x):` x is rounded down to its nearest integer. This integer is returned as a double value.
- `rint(double x):` x is rounded to its nearest integer. If x is equally close to two integers, the even one is returned as a double.
- `round(float x):` Returns `(int)Math.floor(x+0.5)`

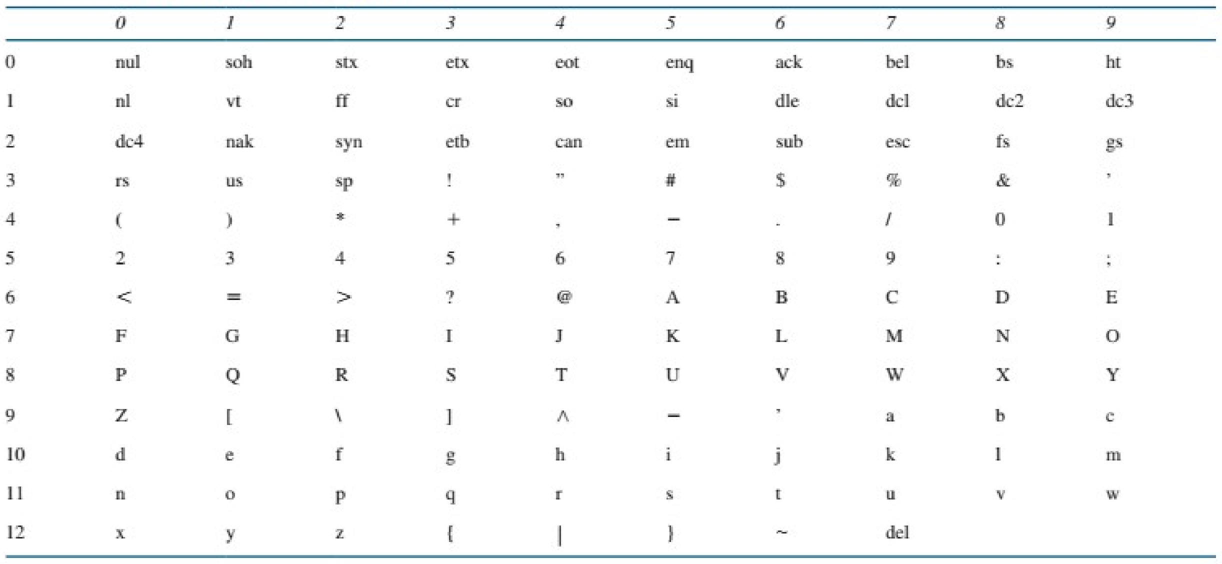
TABLE B.1 ASCII Character Set in the Decimal Index

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	”	#	$	%	&	’
4	(	)	*	+	,	−	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	−	’	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		