First Name: _____ **Last Name**: _____

McGill ID: _____ **Section**: _____

Faculty of Science
COMP-202A - Foundations of Computing (Winter 2017) - All Sections
Midterm Examination

March 8th, 2017
18:00 - 21:00

Examiners: Bentley James Oakes [Section 1 (TR 13:05-14:25)]
Melanie Lyman-Abramovitch [Section 2 (TR 14:35-15:55)]
Giulia Alberini [Section 3 (MWF 9:35-10:25)]

Instructions:

- **DO NOT TURN THIS PAGE UNTIL INSTRUCTED**

- This is a **closed book** examination; only a legal-sized (8.5" by 14") **crib sheet** is permitted. This crib sheet can be single or double-sided; it can be handwritten or typed. Non-electronic translation dictionaries are permitted, but instructors and invigilators reserve the right to inspect them at any time during the examination.
- Besides the above, only writing implements (pens, pencils, erasers, pencil sharpeners, etc.) are allowed. The possession of any other tools or devices is prohibited.
- Answer **all** questions on the answer sheet.
- This examination has **??** pages including this cover page, and is printed on both sides of the paper. On page **??**, you will find information about **useful classes and methods**. **You may detach the Appendix (page ?? onwards) from the examination if you wish.**

Scoring

The exam will be scored as follows:

1. Questions 1 to 8 are worth 1 point each
2. Questions 9 to 25 are worth 2 points each
3. Question 26 is worth 38 points
4. Total: 80 points

Version-0

True/False Section (1 point each)

1. An if-elseif-elseif... sequence must *always* end in an else block.
2. What is the value of z when x and y are both `int`-type variables with value 3?

```
boolean z = (x/2)>(y/3) && !(x<y);
```

3. The following is a valid way of printing the numbers 1 to 10 (inclusive).

```
int x = 10;
while(x > 0){
  System.out.println(11 - x);
  x--;
}
System.out.println(x);
```

4. The same variable name can exist in two different methods.
5. You can compile a class without a `main` method.
6. Every method must have at least one return statement.
7. It is possible to write a for-loop that runs forever.
8. The code below does not compile because you cannot change the length of an array after it has been created.

```
int[] a = new int[5];
int[] b = new int[8];
a = b;
```

Short Answer & Multiple Choice (2 points each)

9. Write 14 (decimal) as a binary number.

10. Write 01101 (binary) as a decimal number

11. What prints?

```
public static void main(String[] args){
  double x = 1/3;
  System.out.println(1 + " is " + x + x + x);
}
```

12. List the names of all variables in scope on the line marked *******HERE******?

```
public static void apple(int x){
  int y = 5;
  if(x<y){
    String s = "banana";
    if(x>s.length()){
      int z = 5;
      z++;
    }
*******HERE******
  }
}
```

13. Which of the following instructions will compile?
Write the letters corresponding to all correct answers on your answer sheet.

A `int x = (int) -3.0/2.0;`

B `int x = 4.5;`

C `double y = 4;`

D `double y = (double) 3/2;`

14. What prints when the following method executes?

```
public static void method(){
  int x = 5;
  int y = 6;
  int z = 7;
  if(x<z){
     System.out.print("A ");
     z = 4;
  }
  if(z>y){
     x = 2;
     System.out.print("B ");
  } else{
    System.out.print(z>x);
  }
}
```

15. Which of the following statements (or group of statements) will cause compile-time errors? Write the letters corresponding to *all* correct answers on your answer sheet.

A
```
String s = Integer.parseInt("twenty");
```

B
```
int result = 35/0;
```

C
```
public static void main (String[] args) {
    String day = "Tuesday";
    today();
}
public static void today() {
    System.out.println("Today is " + day);
}
```

D
```
int[][] a = new int[3][];
System.out.println(a[0]);
```

E
```
int[] a = {1,2,3};
int x = a[-1];
```

16. Write a method header using the following description. Do not write the whole method!

    The name of the method is `isSumEven`. This method will loop through the input parameter `arr`, which is an array of doubles. The method will sum up the values in the array, and returns true if the sum is even, otherwise it returns false.

17. Which of the following are valid ways to call the method whose header is

```
public static double thingOne(int x, double y)
```

Write the letters corresponding to *all* correct answers on your answer sheet.

A
```
int a = 5;
a = thingOne(a,a);
```

B
```
System.out.println(thingOne(1, 4.5));
```

C
```
double b = 5.0;
b = thingOne(b,b);
```

D
```
double b = thingOne(Math.sqrt(16.0), 5.5);
```

E
```
double x = 1;
int y = 2;
double z = thingOne(x,y);
```

18. What prints when the following code is run?

```
public class SomeClass{
  public static void main(String[] args){
    int x = 5;
    int y = 7;
    thingTwo(x,y);
    System.out.println(x + " " + y);
  }
  public static int thingTwo(int x, int y){
    int temp = x;
    x = y;
    y = x;
    x++;
    y++;
    return y;
  }
}
```

19. What prints when the following code is run?

```
public class SomeClass{
  public static void main(String[] args){
    int z = one(2);
    System.out.println(z);
  }
  public static int one(int x){
     return two(two(two(two(three(x)))));
  }

  public static int two(int y){
     if(y%2==0){
        return y;
     }
     return y/2;
  }

  public static int three(int z){
     return z+3;
  }
}
```

20. Assume the following method call is valid.

```
int x = someMethod(5, "7", 'M');
```

Which of the following are possible headers for this method?
Write the letters corresponding to *all* correct answers on your answer sheet.

A `public static int someMethod(int val, String val, char val)`

B `public static int someMethod(double a, int b, char c)`

C `public static int someMethod(int x, String y, int z)`

D `public static double someMethod(int n1, String n2, char n3)`

E `public static double someMethod(int a1, char b1, String c1)`

F `public static int someMethod(double p, String q, char r)`

21. How many times does the letter *A* print when the following code executes?

```
for(int i=0; i<5; i++){
  System.out.println("A");
  for(int j=0; j<10; j+=2){
    System.out.println("A");
    for(int k=10; k>0; k--){
      System.out.println("A");
    }
  }
  System.out.println("A");
}
```

22. What prints when the following code is run?

```
String s = "Hello World!";
for (int i = 1; i < s.length(); i+= 2)
{
        char c = s.charAt(i);
        System.out.print(c);
}
```

23. What are the elements of `a` after the following code executes?

```
int[] a= {1,2,3,4,5,6};
for(int i=0; i<a.length; i++){
  a[i] = a[i%(a.length/2)]+1;
}
```

24. What are the elements of `a` after the following code executes?

```
public static void main(String[] args){
  int[] a = {1,2,3,4};
  int[] b = a;
  method(b);
}
public static void method(int[] b){
  for(int i=0; i<b.length; i++){
    b[i] += i;
  }
}
```

25. What prints? (We are not interested in the specific formatting of deepToString)

```
public static void main(String[] args){
  int[] a = {1,2,3};
  int[] b = {9,8,7};
  int[][] c = {a,b};
  method(c);
  System.out.println(Arrays.deepToString(c));
}
public static void method(int[][] a){
  int temp = a[0][1];
  a[0][1] = a[1][2];
  a[1][2] = temp;
}
```

Long Answer Question (38 points total)

You will write three methods for a program that scores multiple `Strings` to find the `String` with the best score.

Part A: scoreChar Method

Write the `scoreChar` method. This method takes a character and returns the character's score as an integer. A character's score is determined by the rules in the following table:

Character	Score
b, *m*, or *g* (lowercase only)	5
Any digit (0,1...9)	10
A space	-10
Anything else	0

Hint: To determine if a character is a digit, you can use $<$ and $>$ to compare characters against other characters.

Part B: scoreString Method

Write the `scoreString` method. This method takes a `String` and returns a score for the `String` as an integer. This method must not print anything.

The score of a `String` is calculated as:

- The sum of all the scores for each of the characters in the String
  - You must use the `scoreChar` method to do this.
- Add 30 whenever there are two of the same characters in a row
  - For example, *bookkeeper* has three pairs of letters, so you would add 90 to the score
  - Note that the same character might appear more than twice in a row.
    - For example, *caaat* has two pairs of the letter *a*
- Finally, add the length of the `String` to the score

For example, `scoreString("belloo 123")` would return 95.

Here's the calculation: The *b* is 5 points. The *1*, *2*, and *3*, give 10 points each for a total of 30. The space is -10. There are two pairs of double letters (*l* and *o*) to give 60 points. Finally we add 10 points for the length of the String.

Part C: findMaxString Method

Write a method `findMaxString`. This method takes as input an array of `Strings` and returns the `String` with the highest score. Note that this method must use the `scoreString` method to score each `String`.

Assume that the input array holds at least one `String`. If there is a tie in scores, return the `String` with that score which is earliest in the array.

SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

- `Arrays` (package `java.util.Arrays` Methods:
  - `public static boolean equals(int[] a, int[] b)`: Returns true if the two specified arrays are equal to one another.
  - `public static boolean deepEquals(int[][] a, int[][] b)`: Returns true if the two specified arrays are deeply equal to one another.
  - `public static String deepToString(Object[] a)`: Returns a string representation of the "deep contents" of the specified array.
  - `public static String toString(Object[] a)`: Returns a string representation of the contents the specified array.
- `String` (package `java.lang`) Methods:
  - `public boolean equals(Object anObject)`: Compares this `String` to `anObject`.
  - `public int length()`: Calculates the length of this `String`.
  - `public char charAt(int i)`: Gets the `char` at position `i` of the `String`. Note that counting starts from 0 so that to get the first character of the `String` you should input `i` equals 0.
  - `public boolean equalsIgnoreCase(String anotherString)`: Compares, ignoring case considerations, this `String` to `anotherString`.
  - `public int compareTo(String anotherString)`: Compares this `String` to `anotherString` lexicographically; returns a negative value if this `String` occurs before `anotherString`, a positive value if this `String` occurs after `anotherString`, and `0` if both `String`s are equal.
  - `public int compareToIgnoreCase(String anotherString)`: Compares, ignoring case considerations, this `String` to `anotherString` lexicographically; returns a negative value if this `String` occurs before `anotherString`, a positive value if this `String` occurs after `anotherString`, and `0` if both `String`s are equal.
  - `public String substring(int start, int finish)`: Returns a new String composed of the `this` String starting from index `start` and up to, but not including index of `finish`
  - `public String replace(char c, char d)` : Returns a new String with all occurrences of the character `c` in the `this` String replaced by the character `d`.
  - `public char[] toCharArray()`: Converts this `String` to a new character array.
- `PrintStream` (package `java.io`) Methods:
  - `public void print(boolean b)`: Prints `boolean` value `b`.
  - `public void print(double d)`: Prints `double` value `d`.
  - `public void print(int i)`: Prints `int` value `i`.
  - `public void print(Object o)`: Prints `Object` `o`.
  - `public void print(String s)`: Prints `String` `s`.
  - `public void println()`: Terminates the current line by writing the line separator string.
  - `public void println(boolean b)`: Prints `boolean` value `b` and then terminates the line.
  - `public void println(double d)`: Prints `double` value `d` and then terminates the line.
  - `public void println(int i)`: Prints `int` value `i` and then terminates the line.
  - `public void println(Object o)`: Prints `Object` `o` and then terminates the line.
  - `public void println(String s)`: Prints `String` `s` and then terminates the line.
- `Math` (package `java.lang`) Methods:
  - `public static double pow(double a, double b)`: Returns the value of `a` raised to the power of `b`.
  - `public static double sqrt(double a)`: Returns the correctly rounded positive square root of `double` value `a`.
  - `public static double random()`: Returns a `double` value with a positive sign, greater than or equal to `0.0` and less than `1.0`.
  - `public static double exp(double a)`: Returns Euler's number e raised to the power of `double` value `a`.

ASCII Table

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	*[SPACE]*	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	*[DEL]*