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

McGill ID: _____ **Section**: _____

Faculty of Science
COMP-202B - Introduction to Computing I (Winter 2012) - All Sections
Midterm Examination

Monday, March 5, 2012
18:00 - 21:00

Examiners: Daniel Pomerantz [Sections 1 and 3]
Xue Liu [Section 2]

Instructions:

- **DO NOT TURN THIS PAGE UNTIL INSTRUCTED**
- This is a **closed book** examination; only a letter-sized (8.5" by 11") **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 this examination paper** and return it. **If you need additional space**, use pages 15-17, and clearly indicate where each question is continued. **In order to receive full marks for a question, you must show all work** unless otherwise stated.
- This examination has **18** pages including this cover page, and is printed on both sides of the paper. On page 18, you will find information about **useful classes and methods**. **You may detach this page from the examination if you wish.**

1	2	3	Subtotal
/4	/8	/8	/20

4	5	6	Subtotal
/6	/24	/6	/36

7	8	9	Subtotal
/8	/22	/14	/44

Total
/100

Section 1 - Short Questions

[4] 1. Given the following java program, please answer question (a) and (b).

```
1 public class GoodLuck {
2     public static void main (String[] args) {
3         System.out.println("Best of luck in your midterm!");
4         System.out.println("Your final mark is determined by +100% preparation and
              -10% unexpected exam paper!");
5     }
6 }
```

(a) Suppose you want to store this java file to your hard drive, what file name should you normally use?

(b) Suppose you are in the directory where this file is stored together with some other java source files. Now you have to compile and run this code by directly using the JDK through typing commands (assume that you can type "javac" to compile .java files and type "java" to run .class files). What command should you type to: (1) compile this code, and (2) run this code after compiling it?

[8] 2. Please judge the correctness of the following statements. If you think a statement is correct, write "**yes**"; otherwise, write "**no**". Please write **ALL** your answers in the table provided below.

Statement	(a)	(b)	(c)	(d)
Correctness				

(a) In java, there are 8 primitive data types, which are byte, short, int, long, float, double, boolean, char.

(b) If a program compiles successfully, then we are sure the program will run from beginning to end without any errors other than incorrect output.

(c) To compare any two Strings and see if they contain the same data value, one can either write "if(S1 == S2)" or "if(S1.equals(S2))".

(d) The following two code segments will print the same results on screen: printing "haha" three times.
Code segment 1:

```
1 int i = 10;
2 while(i > 1); {
3     if(i >= 8) {
4         System.out.println("haha");
5     }
6     i--;
7 }
```

Code segment 2:

```
1  for (int i = 1;;) {
2      if(i >= 8) {
3          System.out.println("haha");
4      }
5      if(i <= 10) {
6          i++;
7      }
8  }
```

[8] 3. For each of the following java expressions (A to D), provide the data type and the value. Please refer to the API reference in appendix, if necessary.

	Expression	Data type	Data value
A	$5 > 6$		
B	"2" + 1 + 0.009		
C	!(5.0 > 6 - 1.0 && !(5.0 ! = 5))		
D	Math.min(6, ("int" + 98).charAt(3))		

Total marks for Section 1: **20**

Section 2 - Long Questions

[6] 4. For-loops and while-loops are syntactically different but functionally equivalent.

(a) Please transform the following for-loop into while-loop.

```
1 for(int i = 0; i < 10; i++) {
2     System.out.println("Good luck!");
3 }
```

(b) Please transform the following while-loop into for-loop.

```
1 int i = 100;
2 Scanner scanner = new Scanner(System.in);
3 while (i > 10) {
4     System.out.println(scanner.nextInt());
5     i--;
6 }
```

[24] 5. The most important thing to do in learning a programming language is to practice coding and perform experiments. Bob is a novice programmer and he loves writing codes to play with different coding constructs. The following is one of Bob's creations.

```
1  public class Interesting {
2  public static void main (String[] args) {
3      final int k = 1;
4      String[] myTestStrings = {'first', 'second'};
5      if(myTestStrings.length() == 2) {
6          //print the second string in myTestStrings to screen
7          System.out.println(myTestStrings[2]);
8      }
9      int[] y;
10     for (int i = 0; i < 100;) {
11         y[i] = k;
12         i = incrementCounter(i);
13     }
14     boolean b = false;
15     if (!b) {
16         System.out.println("This line should always be printed.");
17     }
18     if (b = true) {
19         System.out.println("This line should never be printed in all cases.");
20     }
21 }
22 public static int incrementCounter (int i) {
23     return (i + 1);
24 }
25 }
```

(a) Unfortunately, Bob's code does not compile. We show the compiler errors below. Please help Bob to correct the errors step by step, and make the code compile successfully. For each error message, please explain the error in one or two sentences and then write the exact line of code you would change and what you would change it to to fix the error.

i. Fix the following error.

```
Interesting.java:4: unclosed character literal
String[] myTestStrings = {'first', 'second'};
                          ^
Interesting.java:4: ';' expected
```

Explain:

Line to change	Corrected code

ii. Fix the following error.

```
Interesting.java:5: cannot find symbol
symbol  : method length()
location: class java.lang.String[]
    if(myTestStrings.length() == 2) {
                     ^
```

Explain:

Line to change	Corrected code

iii. Fix the following error.

```
Interesting.java:11: variable y might not have been initialized
        y[i] = k;
        ^
```

Explain:

Line to change	Corrected code

(b) With the corrected code, Bob's program compiles successfully. This time, however, instead of a compilation-time error, a runtime exception occurs. Please explain the error to Bob in one or two sentences and fix this error. Below is the runtime exception Bob receives.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 2
        at Interesting.main(Interesting.java:7)
```

Explain:

Line to change	Corrected code

(c) Finally, with your help, Bob's program run successfully. But Bob is unhappy because line 19 is always executed. Please help to: (1) explain **why**, and (2) **provide programming suggestions** to Bob to avoid this problem.

[6] 6. For the following question, state **what is printed** to the screen by each of the following code segments.

(a) Code segment 1:

```
1  int i = 0;
2  int s = 1;
3  int[] a = {10, 20, 30, 40, 50, 60};
4  for (i = 0; i < a.length; i++) {
5      if (a[i] % 3 == 0) {
6          s = s + (a[i] / s);
7          a[i]++;
8      }
9      System.out.println(s);
10 }
```

The following is printed to screen:

(b) Code segment 2:

```
1  int[] a = {3, 2, 5, 21};
2  int b,c;
3  for (b = 0; b <= 2; b++) {
4      if (a[b] < a[b+1]) {
5          c = a[b];
6          a[b] = a[b+1];
7          a[b+1] = c;
8      }
9  }
10 for (b = 0; b < 4; b++) {
11     System.out.println(a[b]);
12 }
```

The following is printed to screen:

Section 3 - Programming Questions

[8] 7. Write a method "public static String reverse(String str)" to reverse a string. For example, if a string is "m-cgill.ca", after calling reverse("mcgill.ca"), the return result should be "ac.lligcm". There are many different ways to program this method. Here in this question we ***require*** you to program this method according to the following steps:

(a) transform the given string into a char array by: char[] chars = str.toCharArray();

(b) swap the characters in the char array. After swapping characters, the **original char array** should contain characters in the reverse order;

(c) transform the swapped char array back to string and return.

A sketch of the code is given to you below. Please complete the method.

```
1  public static String reverse(String str) {
2
3      char[] chars = str.toCharArray(); //step (a), already completed for you
4
5      //complete step (b) below
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23     return new String(chars); //step (c), already completed for you
24 }
```

[22] 8. In this question you will write a program that "**`Echos`**" the words the user types to screen. We assume that the user only types one word at a time and presses ENTER after typing a word. For example, if the user types "Montreal", your program should print the word "Montreal" to screen.

Additionally, **if** the word is either "**exit**" or "**quit**", your program should terminate without printing the word "exit" or "quit"; **if** the word is "**stat**", your program should print a *two-line* statistical report: the *first line* prints the total number of words the user has entered, and the *second line* prints the total number of alphabetic letters the user has entered. Note that we only count the 26 letters in the alphabet, from a to z, capital letters are counted as lower case letters. For example, if a user has entered 5 words since your program starts, i.e. "We", "love", "java", "a", and "lot!!", the following should be printed to screen after the user types "stat":

```
5
14
```

To implement this program, you need to define a class called "**`Echo`**" and do the following: *(Please read the **entire** question first in order to plan your solution properly)*

(a) write a "**public static void main(String[] args)**" method; add two variables, both of type **int**, to the "main" method to store the total number of words and the total number of letters the user has entered since your program starts;

(b) write a static method "**updateLetterCount**" that takes as input the current count of the letters the user has entered, and the new word the user has just entered. This method updates and **returns the new letter count value**. This method shall be called each time the user enters a new word that is not "exit", "quit", or "stat";

(c) write a static method "**printStat**" that takes as input the total word count and the total letter count, and prints the above-mentioned two-line statistical report on screen, with **no return value**;

(d) create an infinite loop in the "**main**" method; in each iteration of the loop, first ask the user to type a word, and then print the word to screen. The following table gives some example input strings from user and the expected outputs on screen.

Example input	**Expected output**
Montreal	Montreal *(Program waits for new input)*
exit	*(Nothing prints and program terminates)*
quit	*(Nothing prints and program terminates)*
stat	*(Prints the two-line statistical data, and waits for new input)*

Note: Please write your **complete** answer on **next page**.

[14] 9. This programming question has two parts, A and B.

A. Define a class called "Date" which has 3 properties: "private int month", "private int day", "private int year". In addition, you should write the following methods:

(a) provide a getter and a setter for the data field "private int year" (**no need** to write getters and setters for the other two data fields);

(b) write a method called "isLeapYear" which checks if the year contained in the data field "private int year" is a leap year. A leap year is (1) a year that is divisible by 4, i.e., "year % 4 == 0"; (2) every year divisible by 100 is *NOT* a leap year, unless the year is also divisible by 400. This means, for example, year 1800, 1900, 2100, 2200, 2300 and 2500 are *NOT* leap years, while year 2000 and 2400 are leap years;

B. Define a class called "DateComparator" which contains a static method called "compareYear" that compares two Dates for ordering in year. More specifically, "compareYear" takes as input two Date objects and returned "0" if they are in the same year, "-1" if the first Date is in a year ***before*** the second Date, and "1" if the first Date is in a year ***after*** the second Date.

You can assume that the numbers given in the Date objects are valid, i.e., "year" is a positive integer.

Write your answers to part A and part B in the space provided below and on next page.

Total marks for Section 3: **44**

Total marks: **100**

USE THIS PAGE IF YOU NEED ADDITIONAL SPACE. CLEARLY INDICATE WHICH QUESTION(S) YOU ARE ANSWERING HERE.

USE THIS PAGE IF YOU NEED ADDITIONAL SPACE. CLEARLY INDICATE WHICH QUESTION(S) YOU ARE ANSWERING HERE.

USE THIS PAGE IF YOU NEED ADDITIONAL SPACE. CLEARLY INDICATE WHICH QUESTION(S) YOU ARE ANSWERING HERE.

SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

- `String` (package `java.lang`) Methods:
  - `public String(char[] value)`; Constructor, create a new String from a char array.
  - `public boolean equals(String anotherString)`: Compares, this `String` to `anotherString`.
  - `public char[] toCharArray()`: Converts this `String` to a new character array.
  - `public char charAt(int index)`: Returns the character at position `index` in this `String`.
  - `public String replace(char oldChar, char newChar)`: Creates a new `String` in which all occurrences of the character `oldChar` in the original `String` are replaced by the character `newChar`. If the character `oldChar` does not occur in this `String`, the original `String` object is returned.
  - `public int 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`.
  - `public int 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`.
  - `public int length()`: Returns the length of this `String`.
  - `public String substring(int beginIndex)`: Returns a new string that is a substring of this `String` starting from index `beginIndex`.
  - `public String substring(int beginIndex, int endIndex)`: Returns a new string that is a substring of this `String` starting from index `beginIndex` to index `endIndex`.
  - `public String toUpperCase()`: Converts all of the characters in this `String` to upper case.
  - `public String toLowerCase()`: Converts all of the characters in this `String` to lower case.
  - `public String trim()`: Returns a copy of this `String`, with leading and trailing whitespace omitted.
- `Scanner` (package `java.util`) Methods:
  - `public Scanner(InputStream source)`: Constructs a new `Scanner` that produces values scanned from the specified input stream.
  - `public Scanner(File source)`: Constructs a new `Scanner` that produces values scanned from the provided file.
  - `public double nextDouble()`: Scans the next token of the input as a `double`.
  - `public boolean nextBoolean()`: Scans the next token of the input as a `boolean`.
  - `public int nextInt()`: Scans the next token of the input as an `int`.
  - `public String next()`: Scans and returns the next token of the input.
  - `public boolean hasNext()`: Returns true if this scanner has another token in its input.
- `PrintStream` (package `java.io`) Methods:
  - `public void print(char c)`: Prints `char` value `c`.
  - `public void print(char[] s)`: Prints the array of `char` `s`.
  - `public void print(int i)`: Prints `int` value `i`.
  - `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(char c)`: Prints `char` value `c` and then terminates the line.
  - `public void println(char[] s)`: Prints array of `char` `s` 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.
- `Character` (package `java.lang`) Methods:
  - `public static boolean isDigit(char ch)`: Determines if character `ch` is a digit.
  - `public static boolean isLetter(char ch)`: Determines if character `ch` is a letter.
- `Math` (package `java.lang`) Methods:
  - `public static int min(int a, int b)`: Returns the smaller of two int values.
  - `public static int max(int a, int b)`: Returns the greater of two int values.