First Name: ________________________________ **Last Name**: ________________________________

McGill ID: ________________________________ **Section**: _______

Faculty of Science
COMP-202A - Foundations of Computing (Fall 2015) - All Sections
Midterm Examination

November 5th, 2015 Examiners: Melanie Lyman-Abramovitch [Section 1 (MWF 10:35-11:25)]
18:00 - 21:00 Dan Pomerantz [Section 2 TR (13:00-14:30)]
Melanie Lyman-Abramovitch [Section 3 MWF (12:35-13:25)]

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 scantron sheet.
- This examination has **14** pages including this cover page, and is printed on both sides of the paper. On page 14, you will find information about **useful classes and methods**. **You may detach the Appendix (page 14 onwards) from the examination if you wish.**
- The Examination Security Monitor Program detects pairs of students with unusually similar answer patterns on mulitple-choice exams. Data generated by this program can be used as admissible evidence, either to initiate or corroborate an investigation or a charge of cheating under Section 16 of the Code of Student Conduct and Disciplinary Procedures.
- **MAKE SURE TO WRITE YOUR NAME AND STUDENT ID ON THE SCANTRON AS WELL AS TO FILL IN THE BUBBLE PROPERLY AT THE BEGINNING OF THE EXAM. MARKS WILL BE DEDUCED IF INFORMATION IS MISSING.**

Scoring

The exam will be scored as follows:

1. Questions 1 to 10 are worth 1 point each
2. Questions 11 to 40 will be worth 2 points each
3. Question 41 will be worth 30 points

Version-4

True/False Section (1 point each)

1. If you have an if block, followed by an else-if block, you are required to end with an else block.

   (A) FALSE
   (B) TRUE

2. In Java, if `a` and `b` are boolean expressions, $a\&\&b$ and $b\&\&a$ will always evaluate to the same result.

   (A) TRUE
   (B) FALSE

3. It is possible to successfully compile a .java file that does not contain a main method.

   (A) FALSE
   (B) TRUE

4. In Java, proper indentation is enforced by the compiler.

   (A) FALSE
   (B) TRUE

5. The following code compiles:

```
int x = 1;
if(x) {
   System.out.println("Hello world");
}
```

   (A) TRUE
   (B) FALSE

6. In Java, we can compare an `int` value with a `char` value without doing any explicit typecasting.

   (A) FALSE
   (B) TRUE

7. The size of an array in Java can never be changed after it has been created.

   (A) TRUE
   (B) FALSE

8. What is the value of the variable c after the following code executes?

```
boolean a = true;
boolean b = false;
boolean c = !(a || b);
```

   (A) FALSE
   (B) TRUE

9. After the following code executes, it will print 8 to the screen.

```
int[] a = {8,9,10};
int[] b = a;
b[0] = 10;
System.out.println(a[0]);
```

(A) FALSE

(B) TRUE

10. Declaring a variable with the same name twice inside the same method always results in a compiler error.

(A) TRUE

(B) FALSE

Regular multiple choice (2 points each)

For this section, if ‘error’ is not a choice, assume that the code compiles and runs without error.

11. Which of the following `boolean` expressions is `true` when *exactly* one of `a` and `b` is `true` and `false` otherwise?

    (A) $(a||b)$ && !(a && b)

    (B) !a && !b

    (C) $(a||b)$ && (a && b)

    (D) $(a||b)$ || (a && b)

    (E) !$(a||b)$ && (a&& b)

12. What does the following snippet of code print?

```
int x = 5;
if(x=5){
   System.out.println("x is five");
}else {
   System.out.println("x is not five");
}
```

    (A) There are no errors, but it doesn’t print anything.

    (B) There is a compile-time error.

    (C) x is five

    (D) There is a run-time error.

    (E) x is not five

13. What does the following block of code print?

```
int x = 5;
int y = 6;
int z = 7;
if(x<y){
   System.out.println("cats");
   if(y>z){
      System.out.println("dogs");
      if(z>x){
         System.out.println("rabbits");
      }
   }
}
```

    (A) cats
    dogs
    rabbits

    (B) There is no error, but it prints nothing.

    (C) cats
    rabbits

    (D) cats

    (E) There is a run time error.

14. What does the following block of code print?

```
int x = 5;
int y = 6;
int z = 7;
if(x<y){
   System.out.println("x is smaller than y");
}
if(y<z){
   System.out.println("y is smaller than z");
}
else{
   System.out.println("z is smaller than y");
}
```

(A) There is an error.

(B) z is smaller than y.

(C) x is smaller than y
y is smaller than z

(D) y is smaller than z

(E) x is smaller than y

15. Which of the following are primitive types? Shade in ALL correct answers on your scantron sheet.

(A) String

(B) char

(C) double

(D) boolean

(E) int[]

16. What does the following code print?

```
System.out.println("Hello! " + 4 + (2*7));
```

(A) This code will not run. There is a compile-time error.

(B) Hello! 4(2*7)

(C) Hello! 18

(D) This code will not run. There is a run-time error.

(E) Hello! 414

17. What is the value stored in the variable z?

```
double z = 1/2;
```

(A) 1

(B) 0.0

(C) 0.5

(D) 1.0

(E) Compiler error.

18. What is the value stored in the variable z?

```
int z = 1.0/2.0;
```

(A) 0.0
(B) 1
(C) 0.5
(D) 1.0
(E) Compiler error.

19. What are the values of x, y and z after the following code executes?

```
int x = 5;
int y = 6;
int z = 7;
x = z;
x = x + 1;
z = y;
y = z + x;
z = 2*y;
x--;
```

(A) x=7, y=15, z=30
(B) x=7, y=14, z=12
(C) x=5, y=12, z=24
(D) x=7, y=14, z=28
(E) x=6, y=15, z=24

20. What does the following code print?

```
int x=0;
while(x<10){
   x+=2;
   System.out.print(x + ", ");
}
System.out.println(x);
```

(A) 0, 2, 4, 6, 8, 10, 12
(B) 2, 4, 6, 8, 10
(C) 2, 4, 6, 8, 10, 10
(D) 0, 2, 4, 6, 8, 10
(E) 2, 4, 6, 8, 10, 12

21. What does the following code print?

```
int x=0;
String s = "";
while(x<10){
   x++;
   if(x%3==0){
      x++;
   }
   x++;
   s = s + x + " ";
}

System.out.println(s);
```

(A) 2 5 8 11

(B) 1 4 7 10

(C) 1 4 7

(D) 2 5 8

(E) 1 3 6 9

22. What is the value of the variable z after the code executes?

```
int z = 0;
for (int i = 0; i < 10; i = i + 2) {
     for (int j = 10; j >= 1; j = j - 3) {
          z++;
          j++;
     }

     z++;
}
```

(A) 20

(B) 10

(C) 30

(D) 25

(E) 15

23. What does the following code print?

```
for(int i = 0; i < 10; i + 2){
   System.out.print(i + " ");
}
```

(A) 0 1 2 3 4 5 6 7 8 9

(B) 0 1 2 3 4 5 6 7 8 9 10

(C) 0 2 4 6 8 10

(D) 0 2 4 6 8

(E) There is a compile time error

24. What does the following code print?

```
for(int i=0; i<3; i++){
   for(int j=0; j<i; j++){
      i--;
      System.out.print(j + " ");
   }
}
```

(A) 0 0 0 1 0 1 2 0 1 2 3

(B) This is an infinite loop - it prints forever

(C) 0 0 0 1 0 1 2

(D) 0 0 1 0 1 2

(E) 0 0 1 0 1 2 0 1 2 3

25. What are the contents of the array a after the following code executes?

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

(A) a=[2,3,4,5,5]

(B) There is a run-time error

(C) a=[2,3,4,5,2]

(D) a=[5,4,3,2,1]

(E) a=[2,3,4,5,1]

26. What are the contents of the array a after the following lines of code execute?

```
int[] a = {1,2,3};
method(a[0],a[1],a[2]);

//Below is the entire body of the method called on the last line above.
public static void method(int a, int b, int c){
   int x = a;
   a = b;
   b = c;
   c = x;
}
```

(A) a=[1,2,3]

(B) a=[3,2,1]

(C) a=[3,1,2]

(D) a=[2,1,3]

(E) a=[2,3,1]

27. What are the contents of the array a after the following lines of code execute? Note that the default value when you create an `int[][]` is 0 in each position.

```
int[][] a = new int[5][2];
for(int i=0; i<a.length; i++){
   a[i][i%a[i].length] = i;
}
```

(A) a=[[0, 0],[1, 1],[2, 2],[3, 3],[4, 4]]

(B) a=[[0, 0],[0, 1],[2, 0],[0, 3],[4, 0]]

(C) Compile time error - a.length is undefined for multi-dimensional arrays.

(D) a=[[0,1,2,3,4],[0,1,2,3,4]]

(E) Undefined - not all entries in a have been assigned values.

28. What are the contents of the array a after the following lines of code execute?

```
int[] a = {1,2,3,4,5};
method(a);

//Below is the entire body of the method called on the last line above.
public static void method(int[] a){
   for(int i=0; i<a.length; i++){
      a[i] = a[a.length-i-1];
   }
}
```

(A) a=[1,2,3,4,5]

(B) a=[1,2,3,2,1]

(C) a=[5,4,3,4,5]

(D) Run time off-by-one error

(E) a=[5,4,3,2,1]

29. Consider the following code. What prints?

```
public class Swaps
{
     public static void main(String[] args)
     {
          int[] a = {1,2,3,4,5};
          int[] b = {1,2,3,4};
          swapArrays(a,b);

          int x = 30;
          int y = 40;
          swapInts(3,4);

          System.out.println(a.length + " " + x);
     }

     public static void swapArrays(int[] a1, int[] a2)
     {
          int[] temp = a1;
          a1 = a2;
          a2 = temp;
     }

     public static void swapInts(int i, int j)
     {
         int temp = i;
         i = j;
         j = temp;
     }
}
```

(A) 4 40

(B) Compiler error

(C) 4 30

(D) 5 40

(E) 5 30

30. Which of the following creates an array of integers $[1, 2, 3, 4, 5]$?
Shade in ALL correct answers on your scantron sheet. If the statement results in a run or compile time error, then it is not valid.

(A)
```
int[] badPassword = new int[5];
badPassword = {1, 2, 3, 4, 5};
```

(B)
```
int[] badPassword = new int[5];
for(int i=0; i<=badPassword.length(); i++){
     badPassword[i] = i;
}
```

(C)
```
int[] badPassword = {1,2,3,4,5};
```

(D)
```
int[] badPassword = new int[5];
for(int i=badPassword.length; i>0; i--){
     badPassword[i] = i;
}
```

(E)
```
int[] badPassword = new int[5];
for(int i=1; i<6; i++){
     badPassword[i-1] = i;
}
```

For questions 31 to 34 use the following code. See the appendix for a description of `Arrays.toString()` which is used below to print the array contents.

```
import java.util.Arrays;
public class MidtermCode{
  public static void main(String[] args){
    int[] a = {1,2,3,4,5};
    int[] b = {5,4,3,2,1};
    int[] c = merge(a, b);
    System.out.println(Arrays.toString(c));
  }
  public static int[] merge(int[] a, int[] b){
    int[] c = new int[a.length+b.length];
    for(int i=0; i<c.length; i++){
      if(i>=a.length){
        c[i] = b[i-a.length];
      }
      else{
        c[i] = a[i];
      }
    }
    return c;
  }
}
```

31. What is displayed on the console after the System.out.println() call in the main method?

(A) Nothing, there is run time error

(B) $[1, 2, 3, 4, 5, 5, 4, 3, 2, 1]$

(C) $[5, 4, 3, 2, 1, 1, 2, 3, 4, 5]$

(D) $[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]$

(E) $[5, 4, 3, 2, 1, 5, 4, 3, 2, 1]$

32. If we change $if(i >= a.length)$ to $if(i > a.length)$, what is displayed on the console after the System.out.println() call in the main method?

    (A) $[5, 4, 3, 2, 1, 1, 2, 3, 4, 5]$

    (B) $[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]$

    (C) $[1, 2, 3, 4, 5, 5, 4, 3, 2, 1]$

    (D) Nothing, there is run time error

    (E) $[5, 4, 3, 2, 1, 5, 4, 3, 2, 1]$

33. If we change $c[i] = b[i - a.length]$ to $c[i] = b[i - a.length - 1]$, what is displayed on the console after the System.out.println() call in the main method (assume the change from the previous question was reversed).

    (A) $[5, 4, 3, 2, 1, 5, 4, 3, 2, 1]$

    (B) $[1, 2, 3, 4, 5, 5, 4, 3, 2, 1]$

    (C) $[5, 4, 3, 2, 1, 1, 2, 3, 4, 5]$

    (D) $[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]$

    (E) Nothing, there is run time error

34. If we change $c[i] = b[i - a.length]$ to $c[i] = b[b.length + a.length - i - 1]$, what is displayed on the console after the System.out.println() call in the main method? (assume the change from the previous question was reversed).

    (A) $[1, 2, 3, 4, 5, 5, 4, 3, 2, 1]$

    (B) $[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]$

    (C) Nothing, there is run time error

    (D) $[5, 4, 3, 2, 1, 1, 2, 3, 4, 5]$

    (E) $[5, 4, 3, 2, 1, 5, 4, 3, 2, 1]$

35. Consider the following method signature

    ```
    public static double myMethod(int a, double b)
    ```

    Which of the following are valid ways to call the method?
    Shade in ALL valid answers on your scantron.

    (A) myMethod(3, myMethod(2, 4));

    (B) int x = (int)myMethod(1, 2);

    (C) myMethod(3.0, myMethod(2, 4));

    (D) System.out.println(myMethod(4.0, 3));

    (E) myMethod(3, (double) myMethod(2, 4));

36. Suppose you have a method that returns `double`. Which of the following is NOT a valid return statement inside this method? Note: there is only one correct answer.

    (A) `return (double)"3";`

    (B) `return 3.0;`

    (C) `return (int)3.0;`

    (D) `return '3';`

    (E) `return 3;`

37. Consider the following method signature
`public static double myMethod(int[] a)`

Which of the following are valid ways to call the method. If the call results in a run or compile time error, this is not a valid call. Shade in ALL valid answers on your scantron.

(A)
```
int[][] a = {{1, 2, 3}, {2, 3, 4}};
myMethod(a[a.length]);
```

(B)
```
int[] a = {1, 2, 3, 4, 5};
myMethod(a);
```

(C)
```
double[] a = {1, 2, 3.0, 4.0, 5.0};
myMethod(a);
```

(D)
```
int[][] a = {{1}, {2}};
myMethod(a);
```

(E)
```
int[][] a = {{1}, {2}};
myMethod(a[0]);
```

38. What is the value of the variable y after the following snippet of code executes?

```
double x = 1.5;
double y = (int)x;
x = .5;
```

(A) 2.0

(B) 1.0

(C) 0.0

(D) 1.5

(E) 0.5

39. Which of the following Java expressions does NOT have a value of 0?

(A) `1 / 2;`

(B) `(int).5;`

(C) `(int)(-.5);`

(D) `(double)1/2;`

(E) `(double)(1/2);`

40. What does the following code print?

```
String s = "12345";
for (int i = 0; i < s.length(); i++)
{
     s.charAt(i) = s.charAt((i + 1) % s.length());
}
```

(A) This code will not run. There is a compile-time error.

(B) This code will not run. There is a run-time error.

(C) 23451

(D) 51234

(E) 23452

Long Answer Questions (30 points total)

For this question, you will write several methods for your own Math class. These should all be inside a class called Hypotenuse. **Write these methods on the provided paper. If you run out of space, more paper will be provided. Make sure that your name and student number are written on the extra paper.**

- Write a method called absVal. This method takes as input a real number and outputs the absolute value of that number. For example absVal(3.5) should return 3.5, absVal(-4) should return 4, and absVal(0) should return 0. **You may NOT use any library methods to help for this question.**
- Write a method called power. This method takes as input a real number, x, and an integer number, y. It outputs the real number to the power of the integer number, x^y. You can assume that the exponent (the integer, y) is non-negative (but it might be 0). **You may NOT use any library methods to help for this question.**
- Write a method called squareRoot. This method takes as input a real number, y, and outputs its square root, $\sqrt{y}$. You will do this using the *Babylonian* method described in detail below. **You may NOT use any library methods to help for this question.**

  The *Babylonian* method for approximating the square root of a number works by making successive "guesses" (denoted by $x_0, x_1, ...x_k$ below) which get closer and closer to the actual value. Once the difference between successive guesses is very small, the method stops and the final guess is considered to be the answer.

  - The first guess :$x_0 = \frac{y}{2}$
  - Subsequent guesses: Repeat $x_{k+1} = \frac{1}{2}(x_k + \frac{y}{x_k})$
  - Stop when the difference between successive values of x_k and x_{k+1} is less than 0.1.
  - The answer is the last value of x_{k+1}.

  For this question, if the method input is negative, you should take the absolute value of the input before computing the square root. Do this using the absVal method you wrote in part 1.
- Inside of the class Hypotenuse, write a main method. Your method will ask the user for two floating point numbers as input and obtain them using a Scanner type variable. These numbers correspond to the base and the height of a right angle triangle. It then calculates and prints the length of the hypotenuse. Recall that $c^2 = a^2 + b^2$ where c is the length of the hypotenuse and a and b are the lengths of the base and the height. Use the power and squareRoot methods that you wrote previously to do this.

SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

- `Arrays` (package `java.util.Arrays` Methods:
  - `public int[] copyOfRange(int[] original, int from, int to)`: Returns a subset of the `original` starting at `from` and finishing at `to`, excluding `to`. `to` might lie outside of the 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.
- `File` (package `java.io`) Methods:
  - `public FileSstring pathname)`: Creates a new `File` instance that corresponds to the given pathname.
- `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 f)`: Constructs a new `Scanner` that produces values scanned from the specified 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 nextLine()`: Advances this `Scanner` past the current line and returns the input read.
  - `public boolean hasNextLine()`: Checks whether there are further lines left to scan.
- `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`. (base e) of `double` value `a`. of `double` value `a`.