Faculty of Science
COMP-202C - Foundations of Computing (Summer 2016) - All Sections
Midterm Examination

June 1^{st}, 2015
13:00 - 15:00

Examiners:
Dan Pomerantz

Instructions:

- **DO NOT TURN THIS PAGE UNTIL INSTRUCTED**
- Write all your answers in the answer sheet provided. We will not be collecting the question sheet.
- 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.
- This examination has **6** pages including this cover page, and is printed on both sides of the paper. On page 6, you will find information about **useful classes and methods**. **You may detach the Appendix (page 6 onwards) from the examination if you wish.**
- **MAKE SURE TO WRITE YOUR NAME AND STUDENT ID LEGIBLY ON THE ANSWER SHEET. MARKS WILL BE DEDUCED IF INFORMATION IS MISSING.**

Scoring

The exam will be scored as follows:

1. Questions 1 to 20 are worth 1 point each
2. Questions 21-23 are worth a total of 30 marks

Short Answer (1 point each)

1. What is the type of the following expression? `(double)(1/2)`

2. What is the value of the following expression? `(double)(1/2)`

3. What is the value of the following expression? `13 % 9`

4. Consider the following code: What prints when it is executed?

```
public class MidtermExamples{
  public static void main(String[] args) {
   int x = 10;
   int y = 20;
   y = confusing(y,x);
   System.out.print(x + " " + y + " ");
  }

  public static int confusing(int x , int y) {
   int temp = y;
   y = x;
   x = temp;
   System.out.print(x + " " + y + " ");
   return x;
  }
}
```

5. Consider the following code: What prints when it is executed?

```
public class MidtermExamples{
  public static void main(String[] args) {
    int x = 3;
    increment(x);
    System.out.print(x);
  }

  public static int increment(int x) {
   x++;
   return x;
  }
}
```

6. Suppose you run the following code with the command line argument of 10 (That is, in Dr. Java, you execute the code by typing `run MidtermExamples 10`) What happens?

```
public class MidtermExamples{
  public static void main(String[] args) {
    System.out.println(args[1]);
  }
}
```

7. What prints when the following snippet of code is executed?

```
    int x = 4;
    int y = 3;
    String result = "";
    if (x > 0) {
      result = "11111";
    }
    if (y > 4) {
     result = "22222";
    }
    else {
     result = "33333";
    }
    System.out.println(result);
```

8. Consider a method with the header: `public static boolean method(int x, boolean y)` Is `boolean b = method(10+2,true && 0)` a valid way to call it? Write YES or NO

9. Consider a method with the header: `public static void method(String x)` Is `method((String)10)` a valid way to call it? Write YES or NO

10. Consider a method with the header: `public static int method(String x)` Is `method("" + method("10"))` a valid way to call it? Write YES or NO

11. How many iterations does the following loop have:

```
for (int i = 0; i < 100; i++) {

}
```

12. How many iterations does the following loop have:

```
for (int i = 0; i < 100; i+2) {

}
```

13. How many iterations does the following loop have:

```
int i = 0;
while (i < 100) {
     i = 1000;
     System.out.println(i);
     i = 0;
}
```

14. How many times does the word “hello” get printed when the following snippet of code is executed?

```
int i = 0;
while (i < 10) {
     int j = 0;
     while (j < 5) {
          System.out.println("hello");
          j++;
     }
     i++;
}
```

15. What prints when the following code is executed?

```
public class MidtermExamples{
  public static void main(String[] args) {
   int[] foo = new int[10];
   int[] bar = new int[5];
   swap(foo, bar);
   System.out.println(foo.length);
  }
  public static void swap(int[] a, int[] b) {
   int[] temp = a;
   a = b;
   b = temp;
  }
}
```

16. What prints when the following snippet of code is executed?

```
int[] foo = {1,2,3};
int[] foo2 = foo;
int[] bar = {1,1,1};
foo2[1] = 1;
bar[2] = 3;
System.out.print(foo[1]);
System.out.print(" ");
System.out.println(foo == bar);
```

17. What prints when the following snippet of code is executed?

```
int[] a = {1,2,3};
for (int i = 0; i <= a.length; i++) {
 System.out.print(i + " ");
}
```

18. What is the value of the String referred to by the variable s after the following code executes?

```
String fun = "bcd";
String a = "";
for (int i = 0; i < fun.length(); i++) {
  a = a + (char)(fun.charAt(i) + 3);
}

String s = "";
for (int i = 0; i < a.length(); i++) {
  s = s + (char)(a.charAt(i) + 'A' - 'a');
}
```

19. What values are stored in the array referred to by the variable x after the following code executes?

```
boolean[] x = {true, false, false, false};
for (int i = 1; i < x.length; i++) {
 x[i] = !(x[i - 1] || x[i]);
}
```

20. How many arrays are created in the following code?

```
int[] foo = {1,2,3};
int[] bar = foo;
int[] m = foo;
m = new int[10];
int[] copyOfM = m;
```

Programming section (30 points total)

The following code must all go inside of a class called `Midterm`.

21. Write a method `sum` which takes as input an array of int and returns the sum of the elements in the array. For example, if the array contains `10 2 3`, your method should return 15.

22. Write a method `removeNegatives` which takes as input an array of int and returns a new array in which only the non-negative values are kept. For example, if the original array contains `10 -2 0 4 -1`, your method should return an array of length 3 containing `10 0 4`.

23. Write a program which takes the average of all of the *non-negative* command line arguments and then prints the result. Your program must call the methods you wrote in the previous questions. For example, if someone runs the program by typing `run Midterm 5 6 10 0 -4` into Dr. Java (or `java Midterm 5 6 10 0 -4` from the command line), your program should print `5.25`. You may assume that the person running your program types in integers as command line arguments and also that at least one of the numbers typed at the command prompt is non-negative.

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`.