

McGill University  
 School of Computer Science  
 COMP-202B Introduction to Computing 1  
 Midterm Examination  
 Instructors: Bettina Kemme, Carlton Davis, Jun Wang  
 Tuesday, March 13, 2007 (18:00-20:00)

LAST NAME: \_\_\_\_\_  
 FIRST NAME: \_\_\_\_\_  
 STUDENT NUMBER: \_\_\_\_\_

| <i>Your Section</i>      | <i>Class Section</i> | <i>Instructor and Time</i>      |
|--------------------------|----------------------|---------------------------------|
| <input type="checkbox"/> | Section 1            | Bettina Kemme (MWF 11:30-12:30) |
| <input type="checkbox"/> | Section 2            | Carlton Davis (TTh 10:00-11:30) |
| <input type="checkbox"/> | Section 3            | Jun Wang (MWF 12:30-13:30)      |

**Instructions**

This exam is divided into three parts: multiple choice questions (25 points), short answer questions (30 points), and programming questions (45 points), for a total of 100 points. For each section, we give an estimate of the amount of time you should spend on that section. Follow the directions below exactly.

- Answer all questions on the exam paper in the space provided. Give clear and easily readable answers.
- When giving programs, make sure to use reasonable identifier names, good programming style, and include sufficient comments.
- Make sure that you allow time to attempt each question. If you get stuck on a question, move on to the others and come back to the difficult ones later.
- You may want to remember that in Java expressions have the following associativity and precedence:

| highest, evaluated first | operators | associativity |
|--------------------------|-----------|---------------|
| -----                    | -----     | -----         |
|                          | !         | R to L        |
|                          | * / %     | L to R        |
|                          | + -       | L to R        |
|                          | < <= > >= | L to R        |
|                          | == !=     | L to R        |
|                          | &&        | L to R        |
|                          |           | L to R        |
| v                        | =         | R to L        |
| -----                    | -----     | -----         |
| lowest, evaluated last   |           |               |

- The following methods of the class `String` might be useful:

`char charAt (int index)`

returns the character at the specified `index`.

`int compareTo (String str)`

Returns an integer indicating if this string is lexically before (a negative return value), equal to (a zero return value), or lexically after (a positive return value), the string `str`.

`boolean equals(String str)`

returns true if this string contains the same characters as `str` (including case) and false otherwise.

`boolean equalsIgnoreCase (String str)`

Returns true if this string contains the same characters as `str` (without regard to case) and false otherwise.

`int length()`

Returns the number of characters in the string

`String substring (int offset, int endIndex)`

Returns a new string that is a subset of this string starting at index `offset` and extending through `endIndex - 1`.

`String toLowerCase() (String toUpperCase())`

Returns a new string identical to this string except all uppercase (lowercase) letters are converted to their lowercase (uppercase) equivalent.

This exam has 15 pages (including cover pages).

| MULT | SHRT 1 | SHRT 2 | SHRT 3 | LONG 1 | LONG 2 | TOTAL |
|------|--------|--------|--------|--------|--------|-------|
|      |        |        |        |        |        |       |
| 25   | 10     | 4      | 16     | 25     | 20     | 100   |

## PART A: Multiple Choice Questions (25 points)

**You must put your answers in the following table. For each question, there is only one possible answer (letter between a and e).**

|    |    |    |    |    |    |    |    |    |     |     |     |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 |
|    |    |    |    |    |    |    |    |    |     |     |     |

Questions 1-11 have 2 points each. Question 12 has 3 points.

**1. Which of the following statements about the components of a computer is FALSE?**

- (a) Each data type requires one Byte of data storage.
- (b) Each main memory location has a unique numeric address.
- (c) A keyboard is an input device.
- (d) The CPU can directly access the data in main memory.
- (e) A program tells the CPU how to manipulate and/or move information.

**2. Which of the following statements is FALSE?**

- (a) The .class file contains Java Bytecode.
- (b) The .class file contains code written in the machine language for the Intel processor.
- (c) The .java files contains what the programmer types in.
- (d) The Java compiler produces Bytecode.
- (d) None of the above.

**3. In which of the following cases will you get a COMPILE-TIME error (an error when you try to compile the program)?**

- 1. You use a variable without declaring it.
  - 2. You declare a variable without using it.
  - 3. You assign to a variable of type double the value of a variable of type int.
  - 4. You assign to a variable of type int the value of a variable of type double.
  - 5. You put a semicolon directly after the for-loop (e.g.: for (i=0; i<n; i++); { ...})
- (a) 1 and 2.
  - (b) 1 and 4.
  - (c) 2 and 4.
  - (d) 3 and 4.
  - (e) 3 and 5.

**4. Which of the following are valid names for variables**

1. `_1stSwitch`
2. `second_Switch`
3. `someSwitch`
4. `switch`
5. `Switch`

- (a) none of them  
(b) 2,3,4  
(c) 1,2,3  
(d) 1,2,3,5  
(e) 2,3,4,5

**5. Consider the following piece of code**

```
int x = 3;  
double y = 9;  
System.out.println(((y+1)/x*2+x*3/y));
```

What does the code print?

- (a) 2  
(b) 2.6666666666666667  
(c) 7.6666666666666667  
(d) 7  
(e)  $((y+1)/x*2+x*3/y)$

**6. Consider the following variable initialization**

```
int a=1;  
boolean b=true;
```

Which of the below statements result in an infinite loop?

1. `while (--a>0) System.out.println("This");`
2. `while (a>0) System.out.println("Question");`
3. `while (b) System.out.println("is");`
4. `while (b==false) System.out.println("Tricky");`

- (a) 1, 2 and 3  
(b) 1 and 2  
(c) 2 and 3  
(d) 2, 3 and 4  
(e) all of above

7. Consider the following parts of a class definition:

```
public class MyClass {  
  
    private static final int aNumber = 6;  
  
    ...  
  
    public void aMethod()  
    {  
        aNumber = aNumber+1;  
        ...  
    }  
  
    ...  
}
```

- (a) These statements result in a compile-time error since `aNumber` is a constant.
- (b) These statements compile without error.
- (c) These statements result in a compile-time error since `aNumber` changes dynamically although it is declared static.
- (d) These statements result in a compile-time error indicating that `final` is an invalid keyword.
- (e) These statements result in a compile-time error indicating that `aNumber` might not be initialized.

8. Which of the following statements is FALSE?

- (a) Methods may change the state of an object (its member variables).
- (b) A method may have multiple parameters but it always has at most one return value.
- (c) Memory for local variables is allocated each time a method is called and discarded when the method finishes execution.
- (d) A method can have a formal parameter that has the same name as a member variable of the class.
- (e) A method can have a formal parameter that has the same name as a local variable of the method.

9. A variable declared `public static varName` in a class `ClassA` with an object `obj` can be accessed from outside the class as follows:

1. `obj.varName`
  2. `ClassA.varName`
  3. Static variables can be accessed directly from outside a class without using an object or the class name.
  4. Static variables cannot be accessed from outside the class.
- (a) Choice 1 only is correct
  - (b) Choice 2 only is correct
  - (c) Choice 3 only is correct
  - (d) Choice 4 only is correct
  - (e) Choices 1 and 2 are correct

10. What is the result of attempting to compile and run the following piece of code?

```
boolean b = false;
int i = 1;
do
{
    i++;
    b = ! b;
} while (b);
System.out.println(i);
```

- (a) A compile-time error because of the line "while(b)": b is an invalid conditional expression for the do-while statement.
- (b) A compile-time error since b=! b is not a valid statement.
- (c) The code compiles and running it will print 1
- (d) The code compiles and running it will print 2
- (e) The code compiles and running it will print 3.

11. Consider the following piece of code

```
int x = 3;
int y = 0;
while ((--x > 0) || ( y > 2) )
{
    System.out.print("*");
    y++;
}
```

How many stars are printed?

- (a) none
- (b) 1
- (c) 2
- (d) 3
- (e) infinite many (program does not terminate)

**12. Consider the following piece of code**

```
public class Test {  
  
    public static int foo(int x, String s)  
    {  
        s = "foo";  
        x = x+1;  
        return x;  
    }  
  
    public static void main(String[] args)  
    {  
        int x=1, y=0;  
        String s= "main";  
        y = foo(x,s);  
        System.out.println("x=" + x + " y=" + y + " s=" +s);  
    }  
}
```

What is the output for the above code?

- (a) x=1 y=0 s=main
- (b) x=2 y=0 s=foo
- (c) x=2 y=2 s=main
- (d) x=1 y=2 s=main
- (e) x=1 y=2 s=foo

## PART B: Short Answer Questions (30 Points)

### 1. (10 points) Very short questions:

- (a) What is the value of the following expression?

`3 / 2 * 2.0 + 1`

ANSWER:

- (b) What is the value of the following boolean expression

`(1 < 2) && ((2 >= 4) || (1 >= 1))`

ANSWER:

- (c) Write a boolean expression that evaluates to `true` if and only if the value of the integer variable `a` is larger than the values of both integer variables `b` and `c`.

ANSWER:

- (d) Write a boolean expression that evaluates to `true` if and only if at most of one of the two integer variables `a` and `b` is 0.

ANSWER:

2. (4 points) What does the following correct Java program print?

```
import java.util.Scanner;
public class MagicPrinter
{
    public static void main (String[] args)
    {
        final int LIMIT = 5;

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

ANSWER:

3. (16 points) The following program takes a string as input. It expects the string only to contain the letters "A", "B", "C", and "F" which represent the grades of students. It is supposed to print for each letter it finds whether the grade is a pass or fail.

```
import java.util.Scanner;
// prints pass and fail grades
public class PrintPassFail {

    public static void main(String[] args)
    {
        String gradeList="";    // the input with all the grades
        char grade;              // each individual grade in the list

        Scanner scan = new Scanner(System.in);

        // get each grade individually
        for (int count = 1; count < gradeList.length(); count++);
        {
            grade = gradeList.charAt(count);
            // print depending on the grade
            switch(grade)
            {
                case 'A':
                case 'B':
                case 'C':
                    System.out.println("pass");
                case 'F':
                    System.out.println("fail");
                    break;
            }
            count++;
        }
    }
}
```

- (a) (6 Points) This program has no compilation errors but semantic errors. Explain each error and indicate why it occurs.
- (b) (8 Points) Rewrite the program so that
- The program has no semantic errors
  - The program prints "Wrong letter!" for each letter in the input string that is not an A,B,C or F.
  - The program uses a while-loop instead of a for-loop
- (c) (2 Points) What does your revised program print out when the input string is "BBAG"?

ANSWER:

## PART C: Programming Questions

1. (25 Points) Create a program that counts the occurrences of a certain pattern in a string. The program should perform the following steps
  - It asks the user to enter an arbitrary phrase (string).
  - It asks the user to enter a pattern (also string) with exactly 2 characters. If the user enters a pattern that does not have exactly two characters, then it should print a warning and ask the user again for the pattern. If the user enters a wrong pattern three times in a row, the program should stop.
  - Once a valid pattern is entered, the program counts how many times the pattern occurs in the sentence and prints the number of occurrences to the screen. In any case the program should ignore whether a letter is written in upper or lower case.
  - Now it asks the user whether he/she wants to check another phrase and pattern, or if he/she wants to exit. If the user enters "y" or "Y", the program should again ask for a phrase and a pattern. Otherwise, the program should exit.

Comments:

- You can write this program using a single main method or decompose the problem into smaller methods.
- If you find it difficult checking for a pattern with 2 characters, try a pattern with 1 character first...

An example execution could be as follows:

```
Enter a phrase:
this is my input
Enter a 2-letter pattern:
pat
The pattern must have exactly 2 characters
Please enter again a pattern:
is
The phrase has 2 occurrences of the pattern is
Do you want to check another sentence (y/n)?
Y
Enter a phrase:
This is another input
Enter a 2-letter pattern:
th
The phrase has 2 occurrences of the pattern th
Do you want to check another sentence (y/n)?
n
```

Another example execution could be as follows:

```
Enter a phrase:
this is my input
Enter a 2-letter pattern:
pat
The pattern must have exactly 2 characters.
```

Please enter again a pattern:

p

The pattern must have exactly 2 characters.

Please enter again a pattern:

patty

3 times incorrect input leads to program exit -- good bye

YOUR PROGRAM:

YOUR PROGRAM CONTINUED:

2. (20 Points) Create a class **Professor**. A professor has a name, a position (full, associate or assistant professor), a salary, and a certain number of years experience in his/her current position. Create the following methods.

- A constructor with one formal parameter for each member variable of the class. It initializes all member variables of the class.
- Getter and setter methods for the salary. The setter method should return a boolean indicating the success (or failure) of the operation. If a call tries to set the salary to a value lower than the current salary the method should return false, otherwise it should set the salary to the new value and return true.
- A method `isSimilar(Professor prof)`. It returns true if the object on which the method is invoked has the same position and the same years of experience as the input object.
- A `toString` method that provides pretty printing of the state of the object.

Find below a helper class **University** with a `main` method that uses all the methods that have to be created. It first creates two professors using the constructor. Then it checks whether the two professors are similar. If they are, the salary of the professor with the lower salary is set to the salary of the other professor. Then, information about both profs (calling the `toString` method) is printed.

```
public class University
{
    public static void main (String[] args)
    {
        // create objects of two famous computer scientists
        Professor prof1 = new Professor("Gray James", "full", 20, 99000.0);
        Professor prof2 = new Professor("Frances E. Allen", "full", 20, 80000.0);
        boolean success;

        // if they have same qualification, they should have same salary
        if (prof1.isSimilar(prof2))
        {
            // returns true is the new salary was actually set.
            success = prof1.setSalary(prof2.getSalary());
            if (! success) // first call not successful, so set the other prof's salary
                prof2.setSalary(prof1.getSalary());
        }

        System.out.println(prof1);
        System.out.println(prof2);
    }
}
```

YOUR PROGRAM:

YOUR PROGRAM CONTINUED: