

AUGUSTANA UNIVERSITY COLLEGE

CSC 120 — ABSTRACTION, DESIGN, AND OBJECT-ORIENTED PROGRAMMING

Mid-term Exam — February 13, 2004

NAME: _____

Instructor: J. Mohr

Time allowed: 50 minutes. Closed book - no references allowed.

Write all answers in the space provided for each question.

The point value of each question is indicated in square brackets in the left-hand margin.
Total points = 75.

[15] TERMS — Match each term or concept in the left column with the phrase in the right column which best defines or describes it by writing the letter preceding each definition in the blank preceding each term.

- | | |
|------------------|---|
| ___ accessor | a. Associating multiple methods with the same name. |
| ___ applet | b. A numeric constant that appears in program code without explanation. |
| ___ cast | c. A class that has no mutator methods. |
| ___ cohesion | d. A graphical Java program that requires a viewer or browser to display and evaluate it. |
| ___ flag | e. A value representing the memory location of an object. |
| ___ immutable | f. An operation which forces the conversion of a value from one type to a different type. |
| ___ magic number | g. The set of values that describe an object and that influence how an object reacts to method calls. |
| ___ overloading | h. The expression supplied for a parameter by the caller of a function. |
| ___ precondition | i. An externally observable effect of a function other than the return of a single value. |
| ___ predicate | j. A method that does not change the state of its implicit parameter. |
| ___ reference | k. A method that has no implicit parameter, or a variable defined in a class that has only one value for the whole class. |
| ___ scope | l. A sequence of instructions that is executed repeatedly. |
| ___ side effect | m. A predicate which must be true when a method is called in order for the method to execute correctly. |
| ___ state | n. The portion of a program within which a variable is visible. |
| ___ static | o. The quality of a class for which all features of its public interface are closely related to the single concept that the class represents. |
| | p. A Boolean variable. |
| | q. A method that returns a Boolean value. |

[10] TRUE OR FALSE - Indicate whether each of the following statements is true or false by writing T or F beside each statement below.

- ___ Java is case-sensitive.
- ___ Java requires that every method be located inside a class.
- ___ The Java compiler (`javac`) detects logic errors while it compiles a program.
- ___ Java source code is stored in a file with the extension `'.class'`.
- ___ All classes from the `java.lang` package are automatically imported.
- ___ In a statement such as

```
int partial = total * 0.05;
```

Java will automatically convert the floating-point result of the expression on the right-hand side to an integer before assigning the value to the variable on the left-hand side.
- ___ The name of a constructor is always the same as the class name.
- ___ `String` objects are immutable.
- ___ Statements may be grouped together to form a block statement by enclosing them in square brackets `[]`.
- ___ In Java, a method can never modify the values of actual parameters that are numeric.

[5] EXPRESSIONS — Given the following declarations

```
double x = 12.8;  
double y = -1.5;  
int m = 18;  
int n = 4;
```

what is the value of each of the following expressions?

- a) $x / n + y$
- b) $m / n + m \% n$
- c) $4 * y - n / 5$
- d) `(int) x`
- e) $x + n * y$

[12] MULTIPLE CHOICE — For each question below, circle the letter preceding the single response which best answers or completes the question or statement.

1. Which of the following lines contain(s) a valid Java statement **and** a Java comment?
 - a. `rate = 0.06; // 6%`
 - b. `a = b * /* remainder */ (1 - h);`
 - c. `System.out.print("Hello, /* world */\n");`
 - d. both *a* and *b*
 - e. both *b* and *c*
 - f. both *a* and *c*
 - g. *a, b, and c*

2. Which of the following is **not** a primitive type in Java?
 - a. `boolean`
 - b. `char`
 - c. `double`
 - d. `String`

3. Which one of the following statements is an assignment statement?
 - a. `double total = count * 0.01;`
 - b. `int x;`
 - c. `sum = sum + value;`
 - d. `count++;`

4. Which of the following words is **not** used in declaring a Java constant?
 - a. `const`
 - b. `final`
 - c. `public`
 - d. `static`

5. Which of the following expressions would correctly extract the digit in the tens column of a four-digit number stored in 'value'?
 - a. `value / 100`
 - b. `10 * value % 100`
 - c. `value % 100 / 10`
 - d. `value - value % 10`
 - e. none of the above

6. Assuming we have already executed the following statement
`int len = str.length();`
which of the following expressions correctly extracts the last character from a `String` stored in variable `str`?
 - a. `str.substring(len, 1)`
 - b. `str.substring(len - 1, len)`
 - c. `str.substring(len, len)`
 - d. `str.substring(len, len + 1)`
 - e. none of the above

7. Which of the following expressions does **not** use the string concatenation operator?
- `"5" + "2"`
 - `"" + 3`
 - `4 + " " + 5`
 - `3 + 8.2`
8. Assuming we have already executed the following statement
`String name = "Sam";`
which of the following expressions changes the contents of the string object 'name'?
- `name + " Smith"`
 - `name.toUpperCase()`
 - `name.substring(0, 1)`
 - `name.length()`
 - none of the above**
9. Which one of the following statements includes a method call?
- `Rectangle wall = new Rectangle(5, 10, 20, 30);`
 - `Rectangle floor = new Rectangle();`
 - `new BankAccount(account1.getBalance());`
 - `BankAccount account2 = account1;`
 - `BankAccount account2 = null;`
10. Assuming we have already executed the statement
`BankAccount acct1 = new BankAccount(1000);`
which of the following statements creates a copy of an object reference?
- `BankAccount acct2 = new BankAccount(acct1.getBalance());`
 - `BankAccount acct2 = (BankAccount)acct1.clone();`
 - `BankAccount acct2 = acct1;`
 - `BankAccount acct2 = null;`
11. Which of the following is/are relational operators?
- `>=`
 - `!=`
 - `=`
 - `&&`
 - both a and b**
 - both a and c**
 - a, b, and c**
12. Which of the following is/are the correct way(s) to determine if two strings `s` and `t` are equal?
- `if (s == t)`
 - `if (s.equals(t))`
 - `if (s.compareTo(t) == 0)`
 - both a and b**
 - both a and c**
 - both b and c**
 - a, b, and c**

[16] SYNTAX — Correct the syntax error (or errors) in each of the following code excerpts by removing, inserting, or changing one or more characters as required. You may assume that the variable `x` has been correctly declared and initialized.

1.

```
if (x > 5);  
    System.out.print("bigger");
```
2.

```
if (x == 0)  
    return 0;  
else if ( x < 0)  
    return -1;  
else (x > 0)  
    return 1;
```
3.

```
if (x == 0) || (x == 1) return 1;
```
4.

```
return (-0.5 <= x <= 0.5);
```
5.

```
public static double average( int a; int b)  
{  
    return (a + b) / 2.0;  
}
```
6.

```
if (x = 1) x++; else x = x + 2;
```
7.

```
int z = 5.0E-3;
```
8.

```
while x < 10  
    x = x + 2;
```

[4] LOOPS — Rewrite the following `for` loop as the equivalent `while` loop.

```
for (int i = 1; i <= 23; i++ )  
    System.out.println(i);
```

SEMANTICS

- [3] 1. Given the following code segment
- ```
s = 0;
if (x > 0) s++;
if (y > 0) s++;
```
- what is the final value of *s* for each of the following combinations of values of *x* and *y*?
- a. *x* is 0, *y* is 0      *s* is \_\_\_\_\_
- b. *x* is 1, *y* is 0      *s* is \_\_\_\_\_
- c. *x* is 1, *y* is 1      *s* is \_\_\_\_\_
- [3] 2. Given the following code segment
- ```
s = 0;
if (x == 0) if (y==0) s = 1; else s = 2;
```
- what is the final value of *s* for each of the following combinations of values of *x* and *y*?
- a. *x* is 0, *y* is 0 *s* is _____
- b. *x* is 0, *y* is 1 *s* is _____
- c. *x* is 1, *y* is 1 *s* is _____
- [4] 3. Use DeMorgan's law to simplify the following Boolean expressions:
- a. `! (x < 0 || y != 0)`
- b. `! (x == 0 || ! (y >= 0 && z < 0))`
- [3] 4. What output is produced by each of the following code segments?
- a.

```
for (int i = 1; i < 1000; i = i * 2)
    System.out.print(" " + i);
```
- b.

```
System.out.println("Paul \"Sparky\" Johnson");
```