

Instructions: This lab is a “quiz” to review Java concepts covered in 138. You are to answer the questions using the Lab01 quiz on the Curator system. First, start a web browser, and open the page `http://cscurator.emich.edu:8080/curator/` to display the login screen. Click on the 238 section in the list, and login using your `my.emich` login ID and your password — an initial password should have been emailed to your `my.emich` account. (If you have not already changed your password, you should do so soon — you should take responsibility of your privacy by choosing a password that is not easy to guess.)

Once you have logged in, open the quiz named Lab01 by clicking the link that says “take this quiz”. You should see a page that looks like an optical scan test sheet with one line for each question. For each question (1–20), click your answer (sorry, the software to generate this sheet uses letters for the choices, and the Curator uses numbers). When done, click the submit. If you do not finish before the lab period is up, you should still submit.

The system will not grade your quiz, but simply collects your answers. I will “mark” your answers and give you feedback by the end of the week. However, you only receive a participation grade for the lab, and so I will not record your grade. This is an exercise to help you remember things you might have forgotten. The correct answers will be provided with your “corrected” submission.

1. Which of the following allows a test whether two references of a class are aliases?

- (a) =
- (b) ==
- (c) an `equals` method of the class
- (d) (a) and (b) only
- (e) (a) and (c) only
- (f) (b) and (c) only
- (g) (a), (b) and (c)
- (h) None of these

2. Which of the following allows a test whether two objects of a class are identical?

- (a) =
- (b) ==
- (c) an `equals` method of the class
- (d) (a) and (b) only
- (e) (a) and (c) only
- (f) (b) and (c) only
- (g) (a), (b) and (c)
- (h) None of these

3. Suppose that the declaration of the class `A` includes a method with the following prototype:

```
boolean lessThan(A rhs)
```

Which of the following tests in the client code correctly compares to objects named `Alpha` and `Beta` using this method?

- (a) `if (Alpha < Beta)`
- (b) `if (Alpha.lessThan(Beta))`
- (c) `if (lessThan(Alpha,Beta))`
- (d) `if (Alpha.lessThan.Beta)`
- (e) `if (lessThan(Alpha).Beta)`
- (f) None of these

For the next three questions assume the following class declaration:

```
class CreditCard {  
    public CreditCard(double amount) { balance = amount; }  
    public void payment(double amount) { balance -= amount; }  
    public void charge(double amount) { balance += amount; }  
    public double cardBalance() { return balance; }  
    private double balance;  
}
```

4. Which method allows observing the value of `balance` in a `CreditCard` object?
 - (a) `payment`
 - (b) `charge`
 - (c) `cardBalance`
 - (d) Choices (a) and (b) only
 - (e) Choices (a), (b), and (c)
 - (f) None of these

5. Which member function allows the value of a `CreditCard` object to be mutated?
 - (a) `payment`
 - (b) `charge`
 - (c) `cardBalance`
 - (d) Choices (a) and (b) only
 - (e) Choices (a), (b), and (c)
 - (f) None of these

6. Consider the following statements

```
CreditCard westernUnion = new CreditCard(1000.0);  
CreditCard bankAmerica = new CreditCard(0.0);  
westernUnion.payment(400.0);           // line 1  
System.out.print(westernUnion.balance); //      2  
bankAmerica = westernUnion;           //      3  
CreditCard.charge(120.0);              //      4
```

Which statements cause compilation errors (select only one):

- (a) 1 only
- (b) 2 only
- (c) 3 only
- (d) 4 only
- (e) 1 and 3 only
- (f) 1 and 4 only
- (g) 2 and 3 only
- (h) 2 and 4 only
- (i) 2, 3 and 4 only
- (j) None of these

7. Which of the following statements about Java classes is *false*?
- (a) Classes can have private methods
 - (b) Classes can have public and private methods and fields
 - (c) Classes can have public data fields
 - (d) Assignment of one object to another copies all of the data fields in aggregate
 - (e) All of these statements are true
8. If the designer of a Java class wishes to allow clients to inspect but not modify private data, what is the best approach?
- (a) Provide an accessor method as a class member.
 - (b) Provide a mutator/modification method as a class member.
 - (c) Declare the data to be public, not private.
 - (d) Provide an additional class constructor.
 - (e) Do nothing because it is not acceptable to let clients inspect private data.
 - (f) None of these

Consider the following class:

```
public class B {  
    private int s;  
    public B(int initS) { s = initS; }  
}
```

9. Assuming everything necessary is in scope, consider the declaration `B foo`; What is the value of `foo.s`?
- (a) 0
 - (b) Unknown
 - (c) The declaration isn't allowed.
 - (d) None of these.
10. Assuming everything necessary is in scope, consider the code fragment:
- ```
B foo = new B(17), bar = new B(32);
bar = foo;
```
- What is the value of `bar.s` after the assignment?
- (a) 17
  - (b) 32
  - (c) Unknown
  - (d) The assignment isn't allowed
  - (e) None of these

For the following 6 questions, consider the class definition:

```
public class Farey {
 private int top, bottom;
 public Farey() { top = bottom = 0; }
 public Farey(int t, int b) { top = t; bottom = b; }
 public Farey add(Farey rhs) {
 return new Farey(top+rhs.top, bottom + rhs.bottom);
 }
 public Farey subtract(Farey rhs) {
 return new Farey(top - rhs.top, bottom - rhs.bottom);
 }
 public boolean equals(Object obj) {
 if (obj instanceof Farey) {
 Farey rhs = (Farey)obj;
 return rhs.top == top && rhs.bottom == bottom;
 }
 else
 return false;
 }
 public void display(PrintStream out) {
 out.print("" + top + "/" + bottom);
 }
}
```

Assuming everything necessary is in scope, consider the following code fragment:

```
Farey a = new Farey(3,5), b = new Farey(1,4), c = new Farey(2,4),
 d = new Farey(0,5), e;
e = a.add(b); //line 1
a.display(System.out); // 2
e = a.add(b).subtract(c); // 3
e = 2*a; // 4
```

11. After the execution of line 1, what are the values of `e.top` and `e.bottom` respectively.
  - (a) 0 and 0
  - (b) 3 and 5
  - (c) 1 and 4
  - (d) 4 and 9
  - (e) Unknown
  - (f) None of these
  
12. What is written to the stream `System.out` when line 2 is executed?
  - (a) "3/5"
  - (b) Nothing
  - (c) None of these

13. After the execution of line 3, what are the values of `e.top` and `e.bottom` respectively?
- (a) 0 and 0
  - (b) 4 and 9
  - (c) 2 and 5
  - (d) The statement isn't allowed.
  - (e) Unknown
  - (f) None of these
14. After the execution of line 4, what are the values of `e.top` and `e.bottom` respectively?
- (a) 6 and 10
  - (b) 6 and 5
  - (c) 3 and 10
  - (d) The statement isn't allowed.
  - (e) Unknown
  - (f) None of these

Consider the following code fragment:

```
Farey x = new Farey(1,2), y = new Farey(2,4);
```

```
if (x.equals(y)) //line 5
 System.out.println("x == y");
else
 System.out.println("x != y");
```

```
if (x.add(x).equals(y)) //line 6
 System.out.println("x + x == y");
else
 System.out.println("x + x != y");
```

15. When the `if` statement beginning on line 5 is executed, what is printed?
- (a) "x == y"
  - (b) "x != y"
16. When the `if` statement beginning on line 6 is executed, what is printed?
- (a) "x + x == y"
  - (b) "x + x != y"

For the following two questions, consider the class:

```
public class Quadratic {
 private double[] coefficient;
 public Quadratic() {
 coefficient = new double[3];
 int i = 0;
 while (i < coefficient.length) {
 coefficient[i] = 0.0;
 i++;
 }
 }
 public Quadratic(double a, double b, double c) {
 coefficient = new double[3];
 coefficient[0] = a;
 coefficient[1] = b;
 coefficient[2] = c;
 }
 double evaluate(double x) {
 return (coefficient[0]*x*x + coefficient[1]*x + coefficient[2]);
 }
}
```

17. Given the declaration `Quadratic f = new Quadratic(1,2,3)`; what value is output by the statement `System.out.print(f.evaluate(2))`;
- (a) 6
  - (b) 11
  - (c) 17
  - (d) None of these
18. Given the declaration `Quadratic g = new Quadratic(1,0,0)`; what value is output by the statement `System.out.print(g.evaluate(2))`;
- (a) 1
  - (b) 4
  - (c) Not allowed
  - (d) None of these

A designer wants to add an addition method to the class `Quadratic`. Consider the partial implementation:

```
----- add(Quadratic rhs) { //line 1
 double a = coefficient[0] + rhs.coefficient[0];
 double b = coefficient[1] + rhs.coefficient[1];
 double c = coefficient[2] + rhs.coefficient[2];
 return -----; //line 2
}
```

19. How should the blank in line 1 be filled?

- (a) `void`
- (b) `Sum`
- (c) `Quadratic`
- (d) It should be left blank
- (e) None of these

20. How should the blank in line 2 be filled?

- (a) `new Quadratic(a, b, c)`
- (b) `Sum`
- (c) `new Quadratic(c, b, a)`
- (d) It should be left blank.
- (e) None of these