University of Washington, Tacoma
Institute of Technology
143 Programming Assessment Exam

Instructions: Please *circle* your answers for the multiple-choice questions and write the answers to the programming questions in the space provided on the same page as the question.

1. Which statement about parameters is *false*?

   (A) The scope of parameters is the method in which they are defined.
   (B) Static methods have no implicit parameter `this`.
   (C) Two overloaded methods in the same class must have parameters with different names.
   (D) In Java, parameters of primitive type (int, double, etc.) are passed by value.
   (E) Two different constructors in a given class can have the same number of parameters.

2. The expression

   ```
   (x && !y)
   ```

   is equivalent to which of the following expressions?

   (A) `(x || !y)`
   (B) `(!x || y)`
   (C) `!(!x || y)`
   (D) `(!x && y)`
   (E) `!(!x && y)`

3. Assume that `x`, `y`, and `z` are all `int` variables. Consider the following code segment:

```
if (x == 0) {
    if (y == 1) {
        z += 2;
    }
} else {
    z += 4;
}
System.out.print(z);
```

   What is printed if `x`, `y`, and `z` are all equal to zero before the code segment executes?

   (A) 0
   (B) 1
   (C) 2
   (D) 4
   (E) 6

The following two questions (#4 and #5) refer to the following (incomplete) definition of the `Employee` class. An `Employee` object will represent one employee, including the person's name and identification number.

```
public class Employee {
    /* fields */
    private String name;
    private String idNum;

    /* methods */
    public Employee(String theName, String theIdNum) { … }
    public Employee(String theName) { … }
}
```

4. Which of the following is a correct definition of variable `emp`?

   I. `Employee emp = "John Smith";`
   II. `Employee emp = new Employee ("Ellen Brown");`
   III. `Employee emp = new Employee ("John Smith", "02345695899123615211");`

   (A) I only
   (B) II only
   (C) III only
   (D) I and III
   (E) II and III

5. Each employee's identification number will be a 20-digit integer. Which of the following correctly explains why the field `idNum` is a `String` rather than an `int`?
   (A) Less storage is required for a `String` than for an `int`.
   (B) An `int` is not used because we don't intend to apply mathematical operations to `idNum` and `String` can store longer numbers than int.
   (C) The field `name` is a `String`; therefore `idNum` must be a `String`, too.
   (D) Although the fields `name` and `idNum` can have different types, a constructor that initializes both fields can only be written if they have the same type.
   (E) A method to change an employee's identification number can be implemented more efficiently if the field is a `String`.

6. Under what conditions can a method be *overloaded*; that is, when can two methods with the same name be included in the same class?
   (A) The methods do different things.
   (B) The methods have different numbers or types of parameters.
   (C) The methods have different parameter names.
   (D) The methods have different return types.
   (E) Two methods with the same name can never be included in the same class.

7. Which of the following best describes what a class's constructor should do?
   (A) Test all of the class's methods.
   (B) Initialize the fields of this instance of the class.
   (C) Determine and return the amount of storage needed by the fields of the class.
   (D) Return to free storage all memory used by this instance of the class.
   (E) Print a message informing the user that a new instance of this class has been created.

8. Consider the following class:

```
public class Sphere {

    public static double volume (int r)
    {
        return 4 / 3 * Math.PI * Math.pow(r, 3));
    }
}
```

   Which of the following statements about this code is true?

   (A) The class will not compile because no constructors are defined.
   (B) The class will not compile because `pi` cannot be declared `public`.
   (C) The class will not compile because the `volume` method is declared `static`.
   (D) `Math.pow(r, 3)` cannot be used because `r` is an `int`.
   (E) The class compiles with no errors but the `volume` method returns a significantly smaller value than the expected $\frac{4}{3}\pi r^3$.

9. Consider the following method:

```
// precondition: a != null; a.length > 0
private static void doIt(double[] a)
{
    double temp;

    for (int k = 0; k < a.length / 2; k++)
    {
        temp = a[k];
        a[k] = a[a.length - 1 - k];
        a[a.length - 1 - k] = temp;
    }
}
```

   Which of the following best describes the task performed by this method?

   (A) Sorts an array in ascending order
   (B) Sorts an array in descending order
   (C) Swaps the first and last elements of an array
   (D) Reverses the order of elements in an array
   (E) None of the above tasks is implemented correctly.

10. What will be output by this code segment?

```
for (int i = 5; i > 0; i--)
{
    for (int j = 1; j <= i; j++)
        System.out.print (j*j + " ");
    System.out.println();
}
```

(A)
```
1
1 4
1 4 9
1 4 9 16
1 4 9 16 25
```

(B)
```
1 4 9 16 25
1 4 9 16
1 4 9
1 4
1
```

(C)
```
25 16 9 4 1
25 16 9 4
25 16 9
25 16
25
```

(D)
```
25
25 16
25 16 9
25 16 9 4
25 16 9 4 1
```

(E)
```
1 4 9 16 25
1 4 9 16 25
1 4 9 16 25
1 4 9 16 25
1 4 9 16 25
```

P1.

Write a method named `numberOfCharacters` that accepts an array of `String` objects and a `char` as arguments. Your method should examine each string in the array to discover the number of letters equal to the `char` in the strings, and it should return an `int` value representing the number of occurrences of that letter in all of the strings in the array. For example, if the array has 17 occurrences of the given `char`, your method should return 17.

The following extended example prints `9`, because the number of occurrences of the `char` 'd' is 9:

```
String[] strings = {"hidden", "Java SDK", "DDD",
                    "parameter", "polymorphism",
                    "dictated", "dodged", "cats and dogs"};
int result = numberOfCharacters(strings, 'd');
System.out.println(result);            // 9
```

You may assume the precondition that the array argument, as well as each string element of the array, is not `null`. If the method is passed an empty array that contains no strings, it should return 0.

P2

Write a Java class named `Stock` that keeps track of a person's purchases of shares of a particular company. The price of a share of stock changes frequently, so shares bought at different times are usually purchased at different prices.

Your class must have the following:

- A constructor that takes no arguments. New `Stock` objects have no shares or cost.
- A method `purchase` that can be called on the `Stock` object to inform it that the person has bought some shares of stock at a given price. The method takes two arguments: a *number of shares* as an `int`, and a *price per share* as a `double`. For example, if the person buys 20 shares at $3.50 per share, for $70.00 total, then `purchase(20, 3.50)` would be called on the person's `Stock` object.
- A `getProfit` method that can be called on the `Stock` object to find out how much money the person has made on shares of the stock (market value is greater than total cost). The method takes one argument: the current price per share of the stock, as a `double`. If the person has lost money on the stock (market value is smaller than total cost), the result of `getProfit` is negative.

  The profit is computed by computing in the following way:
  - compute the current *market value* of all shares of the stock (which is the total number of shares purchased so far, times the current price passed in to the method)
  - subtract the *total cost* of all shares of the stock (the total amount of money the person has paid for all shares of stock purchased so far)

  For example, the following code depicts a person making a net profit of $30.00. This is because the person buys a total of 30 shares (20 + 10) for a total of $90.00 cost, ($70.00 + $20.00), but the new current share price is $4.00, meaning that the shares are worth $120.00.

```
Stock myStock = new Stock();
myStock.purchase(20, 3.50);
myStock.purchase(10, 2.00);
double profit = myStock.getProfit(4.00);
System.out.println(profit);                 // 30.0
```