University of Windsor
School of Computer Science
60-212 – Object-Oriented Programming using Java
Fall 2017
Midterm 1 Examination

Thursday October 19, 2017 **Length: 80 Minutes**

Questionnaire and Answer Booklet

Name (please print): _____

Student Number: _____

Instructions

This is a "**closed book**" examination. No reference material, calculators or any electronic equipment is permitted.
Sign your name in the space provided at the bottom of this page.
Please do not detach the pages of this booklet. If necessary, write on the reverse side of the pages.

Questions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**Total**
Total Mark	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**6**	**4**	**3**	**7**	**6**	**14**	**52**
Your Mark																			

Student's Signature: _____

Multiple-choice questions. Every question has 1 mark, and has only one correct/best answer. Selecting two or more answers for a question will receive zero mark.

JUST CIRCLE AROUND THE SELECTED ANSWER.

1. What is the correct order of the steps in the program development process?

   i. Develop and describe the algorithm.
   ii. Translate the algorithm into Java.
   iii. Understand the problem.
   iv. Compile and test the program.
   v. Test the algorithm with different inputs.

   a. iii, i, ii, iv, v
   b. i, ii, iv, v, iii
   c. iii, i, v, ii, iv
   d. i, iii, v, ii, iv

2. In Java, objects within the same class share common _____ .

   a. data
   b. instructions
   c. comments
   d. behavior

3. Which of the set of instructions has compile-time error?

```
a. Rectangle r = new Rectangle();
   System.out.println(r);
b. int i;
   System.out.println(i);
c. int i = 1;
   System.out.println(i);
d. String s = new String();
   System.out.println(s);
```

4. To follow one of the fundamental aspects of object-oriented programming, Encapsulation, the access specifier in the declaration of instance variables should be _____.

    a. protected
    b. class
    c. public
    d. private

5. What is the return type of a constructor?

    a. void
    b. A constructor does not have a return type.
    c. private
    d. public

6. What terminology describes a method of an object that modifies that object's internal data?

    a. public
    b. void
    c. mutator
    d. accessor

7. What do object variables store?

    a. Objects contents
    b. Objects data and methods
    c. object references
    d. classes

8. What is the output of the following code?

```
int num1 = 1;
int num2 = num1;
num2 = num2 + 1;
System.out.println(num1);
```

   a. 1
   b. 2
   c. 11
   d. num1

9. If a method has two parameters, one explicit and one implicit, and a return type of double, then the documentation comments should include:

   a. One @param statement, and one @return statement
   b. Two @param statements, and one @return statement
   c. One @param statement, and no @return statement
   d. Two @param statements, and no @return statement

10. What is the output of the following code?

```
Rectangle r1 = new Rectangle(10,20,30,40);
Rectangle r2 = r1;
r2.setSize(50,60);
System.out.println(r1.getWidth());
```

   a. 30.0
   b. 20.0
   c. 10.0
   d. 50.0

11. By considering the following code snippet, what is the printed output?

```
int x = 25;
double y = 4;
System.out.println(x/y);
```

a. 6.0
b. 6
c. 6.25
d. it will generate a run-time error, because of data type mismatch

12. By considering the following code snippet, what is the printed output?

```
String str1 = "HighTowerPlace";
String str2 = str1.substring(0,4) + " " + str1.substring(9));
System.out.println(str2);
```

a. High Place
b. Tower Place
c. High P
d. HighTowerPlace

13. [6 marks] Rewrite the following expressions as Java expressions:

$(x \bmod y)\ (1 + xy/4)^n$ (x and y are integers)

```
(x % y) * Math.pow((1 + (x * y) / 4.0),n)
```

$v = 4/3 \times \pi \times r^3$ (v and r are doubles)

```
v = (4 * Math.PI * Math.pow(r,3)) / 3
```

Note: You should use a constant value from a Java API library class for π.

14. [4 marks] What are the values of the following Java expressions? Show the intermediate results for each expression. In each line, assume that:

```
double x = 0.5;
double y = -1;
int m = 10;
int n = 4;
```

```
m / n + m % n
```

```
  2   +   2   =  4
```

```
(int) (x * (m+n))
```

```
(int) (0.5 * 14) = (int) 7.0 = 7
```

```
(double) (m / n)
```

```
(double)    2   = 2.0
```

```
x + m * y - (x + n) * y
```

```
0.5 +  -10  -   4.5  * -1 = -9.5 - (-4.5) = -9.5 + 4.5 = -5
```

15. [3 marks] Suppose you want to declare a **constant** for unit conversion of Kilogram to Pound with the value of 2.20462, that **will be used inside only one method**. Define this constant in Java:

```
final double KILO_TO_POUND = 2.20462;
```

Suppose you want to **use this constant in many methods of a class**. Now, **define the constant for the whole class**:

```
public static final double KILO_TO_POUND = 2.20462;
```

Note: Follow the Java convention for defining constants.

16. [7 Marks] There are 8 errors in the program code given below. Note the comment at the top of the program, which explains the problem that the program should solve. There might be more than one error in one line. Inside the provided table, indicate the line number, the type of error, compile-time or run-time, and the correct version for each error you find. As a guide, the first error has already been answered.

```
import java.awt.rectangle;                                          // Line 0
/**
 Constructs a Rectangle object and then computes
 and prints its area, and the expected result
 */

public class AreaTester                                             // Line 1

    public static void main(String[] args)   {                     // Line 2

        Rectangle r1 = Rectangle(10, 20, 30, 25);                   // Line 3

        int area = r1.getWidth() * r1.getX();                       // Line 4

        System.out.println("Circumference: "  area);                // Line 5

        System.out.println("Expected Area: 600");                   // Line 6

    }                                                               // Line 7

}                                                                   // Line 8
```

LINE #	Error Explanation	Error Type	Corrected version of the line
0	r in rectangle should be capitalized	Compile-time	import java.awt.Rectangle;
1	No { for the class definition	C	`public class AreaTester {`
3	No new command for object instantiation	C	… new Rectangle(…);
4	area should be declared as double	C	double area …
4	r1.getX() is not correct	R	… * r1.getHeight();
5	"Circumference: " is wrong prompt	R	… ("Area: " …
5	No + before variable area	C	… + area);
6	Expected result is wrong	R	… 750");

17. [6 marks] Suppose `Bankaccount` class has two instance variables, `name` that stores name of the account holder, and `balance` to keep the account current balance. Implement a public method `printAccountInfo()` for the `BankAccount` class such that it prints the following information of the object:

```
Today is: DD/MM/YYYY
Account Holder Name: XXXXXXXXXXXXXXX
Account Balance:          9999999.99
```

The values for day, month and year will be sent to the `printAccountInfo()` method using three string arguments. Name these parameters as `day`, `month`, and `year` for the `printAccountInfo()` method.

Note: you should use formatted print command to follow the above format. Consider 15 characters for the name and 10 digits (including one decimal point and two digits after the decimal point) for the balance.

```
public void printAccountInfo(String day, String month, String year) {
        System.out.println("Today is: " + day + "/" + month + "/" + year);
        System.out.printf("Account Holder Name: %15s\n", this.name);
        System.out.printf("Account Balance:        %10.2f", this.name);
}
```

18. [14 marks] Suppose you have the following public interface from a class `Car`:

Class name:
`Car`

Instance variables:
```
String brandName
String modelName
int year
int maxSpeed
```

Constructor #1 :
Arguments: `brandName`, `modelName`, `year`

Constructor #2 :
Arguments: `brandName`, `modelName`, `year`, `maxSpeed`

Accessor methods :
```
String getBrandName()        // it returns the brand name of the object.

String getModelName()        // it returns the model name of the object.

int getYear()                // it returns the year of the object.

boolean isFasterThan(Car anotherCar)    // it returns true if the object
                // has higher or equal maximum speed than that of anotherCar,
                // and false otherwise.
```

Mutator methods:
`setMaxSpeed(int maxSpeed)` // it sets the maximum speed of the car object.

Complete the following tester class using the comments provided, that tests the class `Car`, by creating two instance objects of this class, and call their methods using some testing values. For every print command, simply use System.out.println() method.

```
public class CarTester {

  public static void main (String[] args) {

    // Create an instance object of class Car, car1,
    // with initial values: Toyota, Camry, 2016
    Car car1 = new Car("Toyota", "Camry", 2016);

    // Create another instance object of class Car, car2,
    // with initial values: Honda, Civic, 2017, 220
    Car car2 = new Car("Honda", "Civic", 2017, 220);

    // print the brand and model names of car1
    System.out.println("Car 1 brand name is: " +
                       Car1.getBrandName());
    System.out.println("Car 1 model name is: " +
                       Car1.getModelName());

    // print the year of car2
    System.out.println("Car 2 year is: " + car2.getYear());


    // print the expected output for year of car2
    System.out.println("Expected Car 2 year is: 2017");


    // set the maximum speed of car1 to 210
    car1.setMaxSpeed(210);

    // ********* 2 BONUS MARKS *********
    // Figure out the faster car based on their
    // maximum speeds, and print the result.
    // For instance: "car1 can go faster than car2"
    // or: "car1 can't go faster than car2".

    if (car1.isFasterThan(car2))
      System.out.println("car1 can go faster than car2");
    else
      System.out.println("car1 can't go faster than car2");


  }
}
```