

# SE1011: Exam 2 Name:

---

This is a closed-book, closed-computer, etc. exam. You may use one 8.5"x11" sheet of notes, which you will turn in with your exam. Review all questions before you get started. Use the page at the end of the exam for extra work. Write your name on the front of each page. The exam is printed double-sided. Show all work.

1. (15 points) **Complete** a Java program that uses repeated multiplication to compute integer power  $x^y$ . The program should **print** the number to the console using `System.out`. Hint: Check that your program works for  $3^0$  and  $3^1$ .

```
Scanner in = new Scanner(System.in);
System.out.println("Please enter the base: ");
int x = in.nextInt();

System.out.println("Please enter the exponent: ");
int y = in.nextInt();
```

Name: \_\_\_\_\_

2. (15 pts.) **Complete** a Java main method that displays a GUI input box saying “Enter text:” If the user enters anything into the GUI dialog box, the program should display the message “You entered: \_\_\_\_\_”, filling in the blank with what the user entered. Otherwise, (if the user enters nothing or hits **cancel**) the program should re-display the original dialog box. The redisplayed dialog box should behave exactly like the original dialog box. Use `JOptionPane.showInputDialog(...)` for user input (this method returns `null` if the user presses **cancel**) and `JOptionPane.showMessageDialog(null, ...)` for output.

```
public static void main(String[] ignored) {
```

```
}
```

3. [20 points total]
- a. (4 points) **Describe** the difference between how `public` and `private` methods can be used.
  
  - b. (4 points) **Describe** the difference between a class constant and an instance constant when multiple instances of a class exist.
  
  - c. (6 points) **Give an example** of a line of code that uses the `this` keyword. **Describe** what `this` means in your line of code.
  
  - d. (6 points) **Give an example** of a line of code calling a constructor that could be used in a `main` method. **Describe** how your call to the constructor differs from a normal method call.

4. (15 points) **Draw** a UML diagram for the Runner class used in the program below.

```
public static void main(String[] ignored) {
    Runner bob = new Runner("Bob");
    bob.setRecord(5.9);

    System.out.println("Bob's record time for the mile is "+bob.getRecord() +
                       "minutes");
    bob.run();
    System.out.println("Run, Bob, run!");
}
```

5. [20 points total] Consider a variation on the Dragster class that starts out as below, including only one instance variable.

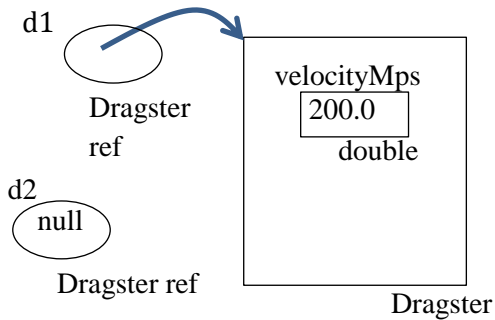
```
public class Dragster {  
    private double velocityMps;
```

- a. (10 points) Implement the `setVelocityMps(double vel)` method of this class:

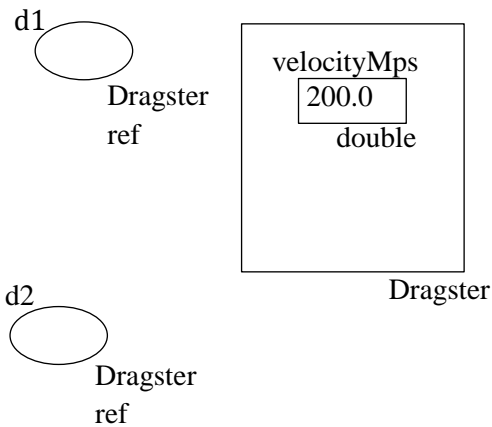
- b. (10 points) Implement the `equals(Dragster other)` method of this class:

6. (20 pts.) The following diagram that illustrates the state of memory at four points in a program on the right.
- Complete** the diagram //a2. Step a has been filled in entirely as an example. The Dragster class is implemented as described in the previous problem.

//a1



//a2 **[COMPLETE THIS]**



```

Dragster d1=null;
Dragster d2=null;
d1 = new Dragster();
d1.setVelocityMps(200);
// a1
d2 = new Dragster();
d2.setVelocityMps(200);
// a2
// b
if(d1 == d2) {
    System.out.println("good");
} else {
    System.out.println("bad");
}
    
```

- Write** what is printed by the code-snippet after the comment “b”. **Explain** your answer.

You may use this space for extra work. Indicate the problem you are working on.