# COMP 1409 – Practice Exam Questions

These questions are similar in style (but not identical) to the types of coding problems on the written portion of the Final Exam.

1. Write a class called Book with the following data and methods:
    - instance variables for author’s last name and title
    - a constructor which takes the last name and the title as parameters
    - the constructor assigns these parameters to the instance variables if they are not null; otherwise an exception with an appropriate message is thrown

```
class Book {
     private String authorLastName;
     private String title;

     public Book(String newAuthorLastName, String newTitle) {

          if(newAuthorLastName != null) {
               authorLastName = newAuthorLastName;
           } else {
               throw new IllegalArgumentException(
                            “Invalid Author Last Name”);
           }

           if(newTitle != null){
               title = newTitle;
           } else {
               throw new IllegalArgumentException(
                            “Invalid Book Title”);
           }
     }
}
```

2. Convert the following if/else statement to a switch statement.

```
if (direction.equals(EAST) || direction.equals(WEST) ||
   direction.equals(NORTH) || direction.equals(SOUTH)) {
   System.out.println(“Valid Direction”);
} else {
   System.out.println(“Invalid Direction”);
}
```

Assume EAST, WEST, NORTH and SOUTH are constants.

```
switch (direction) {

   case EAST:
   case WEST:
   case NORTH:
   case SOUTH:
      System.out.println(“Valid Direction”);
      break;
   default:
      System.out.println(“Invalid Direction”);
      break;

}
```

3. Write a class called Person which has instance variables for date of birth and full name. The datatypes of those two instance variables are Date and Name
   The Person constructor expects a Date and a Name parameter, and stores the values in the instance variables as long as they are not null. Also, in the Person class, write the method getYearOfBirth(), which utilizes the Date class’s getYear() method and the method getDateOfBirth() which returns the date instance variable. The Date class code is shown here.

```
class Date{
        private int year;
        private int month;
        private int day;
        public Date(int newYear, int newMonth, int newDay){
                year= newYear;
                month = newMonth;
                day = newDay;
        }
        public int getYear(){
                return year;
        }
}
```

```
class Person{
        private Date        dateOfBirth;
        private Name        fullName;


        public Person(Date newDateOfBirth, Name newFullName){

                if(newDateOfBirth != null){
                        dateOfBirth = newDateOfBirth;
                }
        if(newFullName!= null){
            fullName = newFullName;
        }
    }

    public int getYearOfBirth(){
        return dateOfBirth.getYear();
    }


    public Date getDateOfBirth(){
        return dateOfBirth;
    }
}
```

4. For the class School below with the instance variable studentNames, write a method called getNumStudentNamesThatContain that returns the number of student names containing the given substring. Use the **contains** method on a String to determine if a student name contains the substring.

   Make sure to do null checks on the elements in the studentNames array before using them.

```
class School
{
        private String[] studentNames;

        public int getNumStudentNamesThatContain(String substring) {

            int i = 0;
            int numNamesThatContain = 0;

            while (i < studentNames.length) {

                if (studentNames[i] != null &&
                studentNames[i].contains(substring))
               {
                  numNamesThatContain++;
               }

               i++;
            }

            return numNamesThatContain;

        }
}
```

5. References

```
class B{
        private String text;

        public B(String newText){
                text = newText;
        }

        public void changeText(String newText){
                text = newText;
        }

        public String getText(){
                return text;
        }
}

class A{
        public A(){
                int c = 5;
                System.out.println(c);                  // question (a)

                B b1 = new B(“something”);
                B b2 = new B(“something”);
                B b3 = b1;
                B b4 = null;
                B b5 = new B(“something different”);

                System.out.println(b1);                 // question (b)
                System.out.println(b1 == b2);           // question (c)

                b3.changeText(“new”);
```

```
          System.out.println(b1.getText());       // question (d)
          System.out.println(b3.getText());       // question (e)

          b1 = b5;
          b3 = b4;
          b2 = b3;                                // question (f)
     }
 }
```

(a) What is printed by the line in question 1(a) of class A on the previous page?

______**5**___________________

(b) What is printed by the line in question 1(b) of class A on the previous page?

_______**an address**__________________

(c) What is printed by the line in question 1(c) of class A on the previous page?

_________**false**________________

(d) What is printed by the line in question 1(d) of class A on the previous page?

___________**new**______________

(e) What is printed by the line in question 1(e) of class A on the previous page?

__________**new**_______________

**(f)** How many B objects exist after the line in question 1(f) (of class A on the previous page) is executed?

__________**one**_______________