**Java Final Exam Essay Questions**

1. Identify the 8 intrinsic data types in Java. Create and present a table that shows the amount of memory each type occupies, and the range of values each type supports. Discuss the 4 ways in which these eight types can be grouped. Also discuss why all other Java data types are objects. What benefits are achieved with this object-based design?

**Type: Size: Range:**

Boolean true or false

Byte 1 byte -128 to 127

Char 2 bytes Two byte Unicode characters used for international alphabets

Short 2 bytes -32,768 to 32,767

Int 4 bytes -2,147,483,648 t0 2,147,483,647

Long 8 bytes -9,223,372,036,854,775,808 to 9,223,372,036,854,775,808

Float 4 bytes +/- 1.4e^-45 to +/- -3.4028235e^38

Double 8 bytes +/- 4.9e^-324 to +/- 1.79769313486231570e^308

Int, Long, Short and Byte are integer types, Float and Doubles are floating-point types and Char is a character type. The last type is a Boolean type. The other java data types are objects simply because every object has a type. Since any entity in your program is considered an object, variables of intrinsic types are objects and the benefits achieved in this design come from the consistency when it comes to creating new objects. In other words, since all other data types are treated as objects, they are also instantiated using similar syntax and the “new” keyword.

1. Discuss the concept of method overloading. How does the Java compiler differentiate one method from another? Be sure you understand why a method return type is not part of the method signature. Why do you suppose Java supports method overloading? What benefit does it offer programmers?

Method overloading is when you create more than one version of the same method, but they contain different sets of parameters. The compiler determines which method to call based on the name of the method and the number, order, and type of the parameters. A method return type is not part of the method signature because its allowed to vary the return type in your overloaded methods, but two methods cannot have the identical signatures and different return types. Java supports method overloading because its required in many object-oriented programs and it offers the programmers to benefit of creating the same methods with different data types.

1. Discuss the topic of two-dimensional arrays in Java. How they are typically declared, instantiated, and initialized? When might they prove to be useful? Is it possible to create Java arrays with more than two dimensions?

Two-dimensional arrays can be thought of as several one-dimensional arrays stacked on top of one another. Two-dimensional arrays are typically declared using 2 sets of brackets and the syntax looks like this: int values[] [];

They are instantiated and initialized similar to the way objects are, using the “new” keyword. It typically looks like this: values = new int[] []; Two-dimensional arrays are very helpful when trying to create a gameboard or chart like structure similar to a tic-tac-toe board. Yes, it is possible to create java arrays with more than two dimensions but doing so can become very complicated.