Howard University
College of Engineering and Architecture
Department of Electrical Engineering & Computer Science

Large Scale Programming
Fall 2023

Prof. Bernard P. Woolfolk

First Exam – 100 pts

October 19, 2023

Name Kyle Nwosu

- **Answer all questions, if you need to guess, then guess!!**
- **Your examination should be uploaded to your git repository**
- **No collaboration of any kind is allowed. All questions should be directed to me.**

Section 1: True/False. Clearly mark your answer. Each question is 1pt. (15 pts.)

1. **F** Java allows a class to extend from multiple classes.

2. **T** A subclass can add behavior that is not present in the superclass.

3. **F** Java arrays can have a variable length, allowing you to resize them dynamically.

4. **F** if Engine extends CarPart, then a variable of type Engine can be assigned a CarPart object.

5. **T** If Engine extends CarPart, then a variable of type CarPart can be assigned an Engine object.

6. **T** Java supports multiple inheritance of classes through interfaces.

7. **F** Java supports multiple constructors with the same signature within a class.

8. **T** Java code is a compiled language that creates binary code for a specific architecture and must be recompiled when moved to another machine.

9. **T** A checked exception represents an error that a program cannot ignore (must either catch or declare).

10. **F** Good object-oriented design promotes maximizing the use of global variable and making little use of encapsulation.

11. **T** Deep inheritance hierarchies should be avoided because they can lead to design complexities and maintenance challenges.

12. **F** Exceptions in Java should be used for error handling and regular control flow.

13. **T** Amazon S3 provides an object storage service that allows you to store and retrieve any amount of data at any time.

14. **T** Amazon EC2 (Elastic Compute Cloud) is a managed service provide by AWS that allows you to run serverless applications.

15. **F** Amazon RDS supports both relational and non-relational databases.

Section 2: Multiple-Choice. Clearly mark your answer. Each question is 1pt. (20 pts)

1. Standard code libraries in Java are organized into
a) Interfaces
b) Classes
c) Objects
d) Packages

2. One of the following is NOT TRUE about Object-Oriented Paradigms (OOPs):
a) OOP is a set of techniques and processes focusing on how to analyze and model a real-world problem and to design a solution.
b) The intended benefits of OOP are to solve the “software” crisis, break complexity into small manageable chunks, and make software maintenance easier.
c) OOP allows reuse of components – plug and play
d) OOP solves the entire problem in one program.

3. Which point is FALSE from the following?
a) A class is an object
b) A class is a template or prototype that defines the composition and the behavior of all objects of certain kinds.
c) A class may have fields of composite types
d) From a class you may instantiate/create an object.

4. A process that involves recognizing and focusing on the important characteristics of a situation or object is known as:
a) Encapsulation
b) Polymorphism
c) Abstraction
d) Inheritance
e) Object persistence

5. An object-oriented design tends to focus on <u>blank</u> to identify objects:
a) Verbs
b) Nouns
c) Encapsulation
d) Inheritance

6. Which of the following is one of the benefits of object-oriented programming?
a) Enables code reuse.
b) More understandable code.
c) More maintainable code.
d) All of the above.

7. Which of the following characteristics of an object-oriented programming language restricts behavior so that an object can only perform actions that are defined for its class?
a) Dynamic Binding
b) Polymorphism
c) Inheritance
d) Encapsulation

8. In Java, if there are one or more constructors for a class then
a) Exactly one of the constructors will be called each time an object of that class is created
b) All of the constructors will be called each time an object of that class is created
c) A destructor must also be written.
d) None of the above, classes cannot have constructors

9. What is true about "has-a" and "is-a" relationships? **(Choose two)**.
a) Instance variables can be used when creating a has-a relationship.
b) Inheritance represents an is-a relationship.
c) Inheritance represents a has-a relationship.
d) Instances must be used when creating a has-a relationship.

10. Which of the following is true about a method inherited into a class C?
a) It must be defined in C before C can be instantiated
b) It must be overridden in C before C can be instantiated
c) It always forces C to become abstract
d) It overrides any method in C with the same name

11. The static variable declared in a class is called
a) Global variable
b) Local Variable
c) Class variable
d) Instance variable

12. Which feature of OOP is exhibited by the function overriding?
a) Polymorphism
b) Encapsulation
c) Abstraction
d) Inheritance

13. Which of the following is an application of the principle of inheritance?
a) An object of class A has a reference to a class B object.
b) Several methods have the same name, but have different signatures.
c) Objects created with new are allocated on the heap.
d) Fields are usually declared private.
e) All classes are ultimately derived from the super class called Object

14. The term signature can be used when describing a method. In this context, a signature is:

a) Defined by the import statement for that class.
b) The number of arguments and the data type of each argument.
c) The return type of method (for example, double or String).
d) The variable names in the argument list.
e) All of the above

15. In the example code fragment shown below, the keyword **abstract**:
public abstract class Test { // . . . more class code }
a) Implies that no object of type Test can ever be created.
b) Makes class Test independent of all other classes, in particular, it is not a subclass of the class Object.
c) Requires the implementation of the toString() method.
d) Ensures that only one object of type Test is ever created.
e) All of the above.

16. Which of the following statements about constructors are correct:
a) A constructor has the same name as the class name.
b) A constructor is responsible for the initialization of an objects' instance fields.
c) Constructor methods have no return type.
d) A class can have several constructors.
e) All of the above.

17. Which of these can be overloaded? (1pt.)
a) methods
b) constructors
c) both a & b
d) classes
e) interfaces

18. Why would a class be declared as abstract?
a) Because it doesn't make logical sense to instantiate it
b) So that it can be used as an interface
c) So that it cannot be inherited from
d) Because it has no abstract

19. Given the code. What is true?

```
public class Room {
        private int roomNr;
        private Date beginDtm;
        private Date endDttm;

        public void book(int roomNr, Date beginDttm, Date endDttm) {
           this.roomNr = roomNr;
           this.beginDtm = beginDttm;
           this.endDttm = endDttm;
        }
}
```

a) the code demonstrates polymorphism
b) the class is fully encapsulated
c) the variable roomNr breaks encapsulation
d) variables beginDttm and endDttm break polymorphism
e) the method book breaks encapsulation

20. A single try block must be followed by which of these?
a) finally
b) catch
c) catch or finally
d) catch and finally

Section #3: Short answers. Express your answer clearly but don't write a dissertation either.
(20 pts.)

1. Describe the principles of encapsulation and abstraction in object-oriented programming. Discuss how these principles contribute to build more maintainable and modular code.
    a. The idea of encapsulation is the act of privatizing objects that are present withing the classes. This hides the information and allows it to only be changed from within the class. Abstraction is the act of simplifying the information with a class only showing the information that is important for the user. Both these ideals together promote a reduction in dependencies, an increase in data hiding and allowing an easier change within the class implementation.

2. Discuss the difference between method overloading and method overriding in Java. Clearly explain when you would use each technique.
    a. Method overloading is when there are multiple versions of the same method within a class but there are different sets of parameters. When you want different ways to perform similar operators this is when it is used. Method Overriding is when a sub class changes the information that it takes from the superclass. The subclasses version of the method will be a specialized version of it. It is used when you want to

3. A program reads a web address and tries to connect to the site. As the application developer of the Java class and method that receives the connection message, would you implement the method to throw a checked or unchecked exception when a connection cannot be established? Please defend your answer.
    a. In the cases that the connection is not established there would be a checked exception because it is something that is common in the case that it is either a server or network problem this condition would be the best point of action. But the place where there would be an unchecked exception is if the error is thrown probably due to student i/o error.

4. Briefly describe the difference between a class and an interface in Java.
    a. What we know as a class is what we call the structured diagram of the or blueprint for creating objects and having the attributes and methods. Its main uses is creating objects. An interface is a blueprint that has a

set of abstract methods that are meant to be implemented by a class.

Section #3: Programming problems. Your solutions should be uploaded to your git repository.
(40 pts.)

1. Create a Java class **Book** with the following attributes: **title**(String), **author**(String), and **year**(int). Override the **toString()** method to display the book's title, author, and year. Also, overrides the **equals()** method to compare two **Book** objects based on their title and author.
   (20 pts.)

   Create class **Driver** to create and test your implementation. The following is a synopsis of the content of your Driver …

   ```
    public class Driver {
       public static void main(String[] args) {
   ```

```
        Book book1 = new Book(“To Kill a MockingBird”, “Harper Lee”, “1960”);
        Book book2 = new Book(“To Kill a MockingBird”, “Harper Lee”, “1960”);
        Book book3 = new Book(“1984”, “George Orwell”, “1960”);
        System.out.println(“book 1 equals book2:” + book1.equals(book2));
        System.out.println(“book 1 equals book3:” + book1.equals(book3));
        System.out.println(“Display of book’s title, author and year:” +
           book1.toString());
    }
}
```

Your solution should use package: org.howard.edu.lsp.midterm.problem1

2. Design a Java program that represents a range of integer (int) values. The **Range** class should allow users to create a range with a specified start and end value. It should provide methods to check if a given value is contained within the range, if two ranges overlap with each and return the size of the **Range**.
   (25 pts.)

   Your solution should include:
   1. A **Range** class that represents a range with start and end values.
   2. A method boolean **isValuesInRange(int value**) that checks whether a given value is contained within the range.
   3. A method boolean **doRangesOverlap(Range otherRange)** that checks if the current range overlaps with another given range.
   4. A method int **size(Range)** that returns the number of integers in the range

Create class Driver to create and test your implementation. The following is synopsis of the content of your Driver ….

```
public class Driver {
   public static void main(String[] args) {
      Range range1 = new Range(5, 10);
      Range range2 = new Range(8, 15);
      int testValue = 7;
      System.out.println(“Is “ + testValue + “ in range? ”  +
         range1.isValueInRange(testValue);
      System.out.println(“Do range1 and range2 overlap?” +
         range1.doRangesOverlap(range2)?;
      System.out.println(“Size of range1: “ + range1.size());
   }
}
```

Your solution should use package: org.howard.edu.lsp.midterm.problem2

Final instructions:

Once completed, your exam should be upload to your git repository:

org.howard.edu.lsp.midterm.exam (put your completed word document here. Do not convert to pdf)
org.howard.edu.lsp.midterm.problem1 (programming assignment 1)
org.howard.edu.lsp.midterm.problem2 (programming assignment 2)

Failure to upload to your git repository will result in a significant penalty.