Howard University
School of Engineering &Architecture
Department of Electrical Engineering & Computer Science

Large Scale/Object-Oriented Programming
Spring 2023

Final Exam

100 pts.

Name : Sanjeev Roka

I declare that I have not collaborated with anyone on this examination

Sanjeev Roka
Type Your Name Here

When completed, the Word document exam should be uploaded to your Github repository. The package should be called howard.edu.lsp.final.document

PLEASE DO NOT CONVERT TO PDF WHEN SUBMITTING.

Section 1: True/False, type or color your answer. (25 pts., 1 pt. each)

1. T F When designing a class, each class should be designed to have multiple goals so that your overall design can have fewer classes.
2. T F Design pattern are a mechanism that enable developer to reuse code in their implementations.
   Reuse is one of the goals.
3. T F Design patterns are not applicable to the design of object-oriented software.
4. T F Information hiding makes program maintenance easier by hiding data and procedure from unaffected parts of the program.
5. T F Software designs are refactored to allow the creation of software that is easier to integrate.
6. T F Inheritance provides a mechanism by which changes to lower level classes can be propagated to all super class quickly.
7. T F Design patterns are best thought of as coding patterns.
   It is not only for coding or object oriented design but for generic use to solve problem.
8. T F Frameworks and design patterns are the same thing as far as designers are concerned.
9. T F Factory pattern can be combined with the Singleton pattern.
   Singelton is often combined with Factory Method.
10. T F Creational design patterns are all about Class and Object composition.
11. T F Structural design patterns are all about class instantiation
12. T F Behavioral design patterns are all about Class's objects communication
13. T F Polymorphism is the mechanism that combines processes and data into a single object.
14. T F Because of potential problems, developers must be aware of the effects of modifications in a superclass and in each of the subclasses that will inherit the modifications.
15. T F High levels of inheritance coupling in a system is good and desirable.
16. T F In Java, the signature of a method is completely specified by the name of the method and the parameters that must be passed to the method.
17. T F Each method in a Java class must have a unique name
18. T F In Java, when an instance of a class, or object, is specified as a parameter to a method, a reference to the said object is passed to the method.
19. T F The return value from a Java method must always match the declared return type.
20. T F The relationship between two objects related by composition cannot be changed at runtime.
21. T F if Engine extends CarPart, then a variable of type Engine can be assigned a CarPart object.
22. T F If Engine extends CarPart, then a variable of type CarPart can be assigned an Engine object.
23. T F A try block can occur without an accompanying catch clause or finally clause.
24. T F When iterating a Java HashSet you are guaranteed to retrieve objects stored in the same order they were inserted.

25. T F A Java HashSet allows duplicate entries in the collection.

Section 2: Multiple Choice, type answer below each question. (40 pts., 1 pt. each)

26. Which of the following option leads to the portability and security of Java?

a) Bytecode is executed by JVM
b) Use of exception handling
c) Dynamic binding between objects
d) Proper encapsulation of classes and objects.

Bytecode is executed by JVM => Making Java portable

Proper encapsulation of classes and objects: Makes more security for Object-Oriented Development and Java is one of them.

27. Suppose the class Undergraduate extends the class Student that extends the class Person.

Given the following variable declarations:

```
Person p = new Person();
Student s = new Student();
Undergraduate ug = new Undergraduate();
```

Which of the following assignments are legal?

I. p = ug;
II. p = new Undergraduate();
III. ug = new Student();
IV. ug = p;
V. s = new Person();

a) III and IV
b) I and IV
c) I and II
d) II, III and V

28. Say that methodA calls methodB, and methodB calls methodC. MethodC might throw a `NumberFormatException`. Can the program be written so that methodA handles the exception?

a) No, the exception must be handled by methodC or its caller, methodB.
b) No, methodC must handle the exception.
c) Yes, if the header for methodC says `. . .throws NumberFormatException`
d) Yes, if the headers for methodC and methodB say `. . .throws NumberFormatException`

29. A finally block is executed:

a) Only when an exception occurs.
b) When the exception is a checked exception.
c) When the exception is an unchecked exception.

d) After a try block.

30. Which of these must follow a single try block?
a) finally
b) catch
c) catch or finally
d) catch and finally
We can either use catch or finally. If we do not use catch we must use finally.

31. Which of the following is true about RuntimeException and its subclasses?
a) If a method throws a RuntimeException, the use of the try/catch block is optional.
b) The File IOException class is a subclass of RuntimeException.
c) In general, handling of RuntimeException should be done at compile time.
d) In general, RuntimeException must be caught with a try/catch block.

32. Cohesion is a qualitative indication of the degree to which a module
a) can be written more compactly.
b) focuses on just one thing.
c) is able to complete its function in a timely manner.
d) is connected to other modules and the outside world.

33. Coupling is a qualitative indication of the degree to which a module
a) can be written more compactly.
b) focuses on just one thing.
c) is able to complete its function in a timely manner.
d) is connected to other modules and the outside world.

34. The root interface of the Java Collection framework hierarchy is
a) Collection
b) Root
c) Collections
d) List/Set

35. Which of the following is true about design patterns? (Choose the best answer).
a) Design patterns represent the best practices used by experienced object-oriented software developers.
b) Design patterns are solutions to general problems that software developers faced during software development.
c) Design patterns are obtained by trial and error by numerous software developers over quite a substantial period.
d) All of the above.

36. You want all the clients using class A to use the same instance of class A when the class is instantiated, what should you do to achieve this goal?
a) Mark class A final
b) Mark class A abstract
c) Apply the Singleton pattern to class A

d) Apply the Proxy pattern to class A

37. You have a class that accepts and returns values in British Imperial units (feet, miles, etc.), but you need to use metric units. The design pattern that would best solve your problem is:

a) Adapter
b) Decorator
c) Delegation
d) Proxy

38. Which of the following describes the Facade pattern correctly.

a) This pattern allows a user to add new functionality to an existing object without altering its structure.
b) This pattern is used when we need to treat a group of objects in a similar way as a single object.
c) This pattern hides the complexities of the system and providers an interface to the client using which the client can access the system.
d) This pattern is primarily used to reduce the number of objects created and to decrease memory footprint and increase performance.

39. Which of the following are concerned with communication between objects?

a) J2EE Design Patterns
b) Behavioral Design Patterns
c) Structural Design Patterns
d) Creational Design Patterns

40. Which of the following is correct about Creational design patterns.

a) These design patterns are specifically concerned with communication between objects.
b) These design patterns provide a way to create objects while hiding the creation logic, rather than instantiating objects directly using new operator.
c) These design patterns concern class and object composition. Concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionalities.
d) None of the above.

41. Which of the following describes the Template pattern correctly?

a) In this pattern, a class behavior changes based on its state.
b) In this pattern, a null object replace check of NULL object instance
c) In this pattern, a class behavior or its algorithm can be changed at run time.
d) In this pattern, an abstract class exposes defined way(s)/template(s) to execute its methods.

42. Which of the following describes the Factory pattern correctly?

a) This pattern creates an object without exposing the creation logic to the client and refers to newly created objects using a common interface.

b) In this pattern, an interface is responsible for creating a factory of related objects without explicitly specifying their classes.
c) This pattern involves a single class that is responsible to create an object while making sure that only a single object is created.
d) This pattern is used when we want to pass data with multiple attributes in one shot from client to server.

43. Which of the following describes the Strategy pattern correctly?
a) In this pattern, a class behavior changes based on its state.
b) In this pattern, a null object replaces check of NULL object instance.
c) In this pattern, a class behavior or its algorithm can be changed at run time.
d) In this pattern, an abstract class exposes defined way(s)/template(s) to execute its methods.

44. This design pattern should be used to access the contents of a collection without exposing its internal representation, to support multiple traversal of a collection, and to provide a uniform interface for traversing different collections.
a) Template method
b) Strategy
c) Iterator
d) Factory method

45. Which design pattern should you use when you want to provide a simple interface to a complex subsystem?
a) Adapter
b) Fascade
c) Abstract Factory
d) Singleton

46. Which design pattern should you use when you want to use an existing class, and its interface does not match the one you need?
a) Adapter
b) Template
c) Fascade
d) Proxy

47. Which of these is not an interface in the Collections Framework?
a) Collection
b) Group
c) Set
d) List

48. Which interface restricts duplicate elements?

a) Set
b) List
c) Map
d) (All of these)

49. Which does NOT implement the Collection interface?
a) List
b) Map
c) Set
d) (None of these)
50. Which provides better performance for the insertion and removal from the middle of a list?
a) Vector
b) ArrayList
c) LinkedList
d) (All of these)

51. An unordered array has a search time complexity of:
a) O(log n)
b) O(n)
c) O(n+1)
d) O(1)

52. An ArrayList has a search time complexity of:
a) O(log n)
b) O(n)
c) O(n+1)
d) O(1)

53. An HashSet has a search time complexity of:
a) O(log n)
b) O(n)
c) O(n+1)
d) O(1)

54. What guarantees type-safety in a collection?
a) Generics
b) Abstract classes
c) Interfaces
d) Collection

55. Which is sorted by natural order?
a) LinkedHashSet
b) TreeSet
c) HashSet
d) None

56. Which of these maintains insertion order?
a) List
b) Set
c) All
d) None

57. Which of these maintain insertion order?
a) TreeSet
b) HashSet
c) LinkedHashSet
d) None

58. Which of these provides a get(int index) method?
a) Map
b) Set
c) List
d) All

59. Which one of the following is true?
a) If an exception is raised in a method f and there is no try block in f that handles the exception, then f exits and the exception propagates to the method that called f.
b) If an exception is raised in a method f and there is no try block in f that handles the exception, this if f is not declared as throwing the exception, the entire program terminates.
c) If an exception is raised in method f and the exception is handled by a catch clause within f, then f resumes executing at the point in the code that the exception was raised.
d) The finally clause of a try block is executed only if an exception is not caught by catch clause of that try block.

60. Which of the following is a correct outline of a method that throws Exception when it finds a problem?
a)
```
public void someMethod () {
  ...
  if ( problem )
    throw new Exception("Useful Message");
  ...
}
```
b)
```
public void someMethod () throws Exception {
  ...
  if ( problem )
    Exception("Useful Message");
  ...
}
```

```
c) public void someMethod () throws Exception {
   ...
   if ( problem )
     throw( new Exception("Useful Message") );
   ...
}
d) public void someMethod () throws Exception {
   ...
   if ( problem )
     throw new Exception("Useful Message");
   ...
}
```

Section 3: Programming Problem (40 pts.)

61. The **SongsDatabase** class keeps tracks of song titles by classifying them according to genre (e.g., Pop, Rock, etc.). The class uses a HashMap to map a genre with a set of songs that belong to such a genre. The set of songs are represented using a HashMap. **(25 pts.)**

```
public class SongsDatabase {
      /* Key is the genre, HashSet contains associated songs */
      private Map<String, HashSet<String>> map =
            new HashMap<String, HashSet<String>>();

      /* Add a song title to a genre */
      public void addSong(String genre, String songTitle) {
            //Code it!!
      }

      /* Return the Set that contains all songs for a genre */
      public Set<String> getSongs(String genre) {
            // Code it!!
      }

      /* Return genre, i.e., jazz, given a song title */
      public String getGenreOfSong(String songTitle) {
            // Code it!!
      }
}

Below is sample that uses your implementation:
Hint: You need some of the below in your JUnit test cases.
…
SongsDatabase db = new SongsDatabase();
db.add("Rap", "Savage");
db.add("Rap", "Gin and Juice");
db.add("Jazz", "Always There");
Set<String> songs = db.getSongs("Rap"); // contains Savage and
                                        // Gin and Juice
System.out.println( db.getGenreOfSong("Savage") );// prints "Rap"
System.out.println( db.getGenreOfSong("Always There") );// prints
"Jazz"
```

Be sure to include a JUnit test case for **each** method in SongsDatabase**. (15 pts.)**

****

Instructions:

Your implementation should be uploaded to your GitHub repository (the same you have been using all semester). The package **is howard.edu.lsp.final.problem** (contains implementation and JUnit test cases). At the least, your implementation should have:

1. SongsDatabase.java (implementation)
2. SongsDatabaseTest.java (JUnit test cases)

Good luck!!