Student Name: _____ Student ID # _____

OU Academic Integrity Pledge
On my honor I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signature: _____ Date: _____

Notes Regarding this Examination

Open Book(s) You may consult any printed textbooks in your immediate possession during the course of this examination.

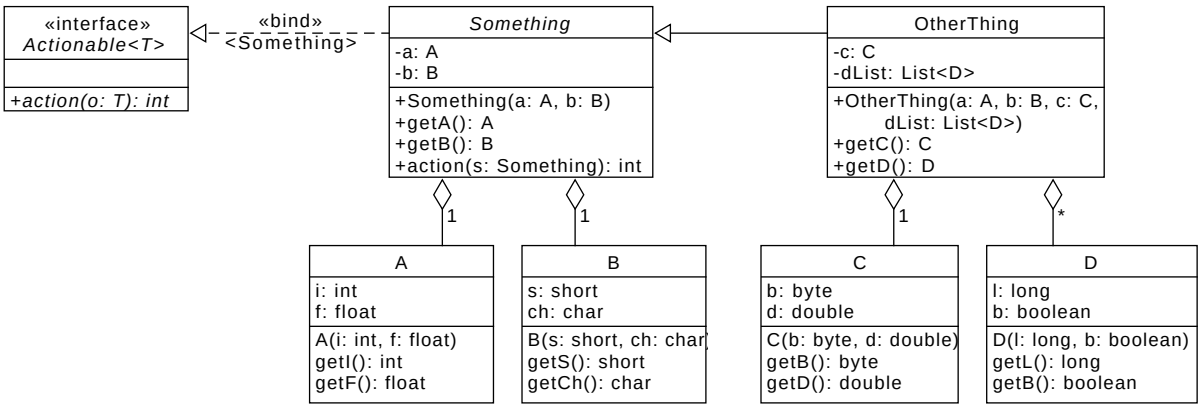
Open Notes You may consult any printed notes in your immediate possession during the course of this examination.

No Electronic Devices Permitted You may not use any electronic devices during the course of this examination, including but not limited to calculators, computers, and cellular phones. All electronic devices in the student's possession must be turned off and placed out of sight (for example, in the student's own pocket or backpack) for the duration of the examination.

Violations Copying another's work, or possession of electronic computing or communication devices in the testing area, is cheating and grounds for penalties in accordance with school policies.

Part I. Object-Oriented Design Components & UML

Refer to the following (partial) UML diagram when answering the questions in this part.



1. (2 points) Which concept is encoded in the UML above?
   - ***A.*** `Actionable` ***is an interface***
   - B. ***`Actionable`*** is an abstract class
   - C. ***`Actionable`*** is a concrete class
   - D. ***`Actionable`*** is a field
   - E. ***`Actionable`*** is a method

2. (2 points) Which concept is encoded in the UML above?
   - A. ***`Something`*** is an interface
   - ***B.*** `Something` ***is an abstract class***
   - C. ***`Something`*** is a concrete class
   - D. ***`Something`*** is a field
   - E. ***`Something`*** is a method

3. (2 points) Which concept is encoded in the UML above?
   - A. `OtherThing` is an interface
   - B. `OtherThing` is an abstract class
   - ***C.*** `OtherThing` ***is a concrete class***
   - D. `OtherThing` is a field
   - E. `OtherThing` is a method

4. (2 points) Which concept is encoded in the UML above?
   - A. `a` is an interface
   - B. `a` is an abstract class
   - C. `a` is a concrete class
   - ***D.*** `a` ***is a field***
   - E. `a` is a method

5. (2 points) Which concept is encoded in the UML above?
   A. `getA()` is an interface
   B. `getA()` is an abstract class
   C. `getA()` is a concrete class
   D. `getA()` is a field
   E. `getA()` is a method

6. (2 points) Which concept is encoded in the UML above?
   A. *`Something`* is `A`
   B. `A` is *`Something`*
   C. `Something` is `Actionable`
   D. *`Actionable`* is *`Something`*
   E. *`Actionable`* is `OtherThing`

7. (2 points) Which concept is encoded in the UML above?
   A. *`Something`* is an `OtherThing`
   B. `OtherThing` is a `Something`
   C. `OtherThing` is a `C`
   D. `C` is an `OtherThing`
   E. `List<D>` is an `OtherThing`

8. (2 points) Which concept is encoded in the UML above?
   A. *`Something`* has an `OtherThing`
   B. `OtherThing` has a *`Something`*
   C. `OtherThing` has a `C`
   D. `C` has an `OtherThing`
   E. `List<D>` has an `OtherThing`

9. (2 points) Which concept is encoded in the UML above?
   A. *`Something`* has any number of `OtherThing`s
   B. `OtherThing` has any number of *`Something`*s
   C. `OtherThing` has any number `D`s
   D. `D` has any number of `OtherThing`s
   E. `List<D>` has any number of `OtherThing`s

10. (2 points) Which concept is encoded in the UML above?
    A. *`Something`*`.action()` uses *`Actionable`*`.action()`
    B. *`Actionable`*`.action()` overloads *`Something`*`.action()`
    C. *`Actionable`*`.action()` overrides *`Something`*`.action()`
    D. *`Something`*`.action()` overloads *`Actionable`*`.action()`
    E. `Something`.`action()` overrides `Actionable`.`action()`

Part II. From Object-Oriented Design to Code

11. (2 points) Which code fragment could be an example of the concept "`W` is a subclass of `X`?"
    A. `X extends W`
    B. `W extends X`
    C. `X implements W`
    D. `W implements X`
    E. `X<W>`

12. (2 points) Which code fragment could be an example of the concept "`X` exhibits the feature `W`?"
    A. `X extends W`
    B. `W extends X`
    C. `X implements W`
    D. `W implements X`
    E. `X<W>`

13. (2 points) Which code fragment could be an example of the concept "`y` is an instance field?"
    A. `int y()`
    B. `static int y()`
    C. `int y`
    D. `static int y`
    E. `class y`

14. (2 points) Which code fragment could be an example of the concept "`y` is a class field?"
    A. `int y()`
    B. `static int y()`
    C. `int y`
    D. `static int y`
    E. `class y`

15. (2 points) Which code fragment could be an example of the concept "`y` is an instance method?"
    A. `int y()`
    B. `static int y()`
    C. `int y`
    D. `static int y`
    E. `class y`

16. (2 points) Which code fragment could be an example of the concept "`y` is a class method?"
    A. `int y()`
    B. `static int y()`
    C. `int y`
    D. `static int y`
    E. `class y`

17. (2 points) Which code fragment could be an example of the concept "`U` has a field of type `V`?"

A. `U<V>`
B. `class U { V v; ... }`
C. `class U { V v(); ... }`
D. `class V { U u; ... }`
E. `class V { U u(); ... }`

18. (2 points) Which code fragment could be an example of the concept "`U` has a method that returns type `V`?"

A. `U<V>`
B. `class U { V v; ... }`
C. `class U { V v(); ... }`
D. `class V { U u; ... }`
E. `class V { U u(); ... }`

19. (2 points) Which code fragment could be an example of the concept "`V` is the generic type of `U`?"

A. `U<V>`
B. `class U { V v; ... }`
C. `class U { V v(); ... }`
D. `class V { U u; ... }`
E. `class V { U u(); ... }`

20. (2 points) Which code fragment could be an example of the concept "`O` has a parameter of type `P`?"

A. `class O(P p) { ... }`
B. `class P(O o) { ... }`
C. `class O() { P p; ... }`
D. `class P() { O o; ... }`
E. `P p; O(p);`

Note that E shows an argument rather than a parameter. However, due to the possibly confusing notation in parts A-D (which would have been better written without the word "class" or with additional ellipses), E is counted for credit on the exam.

21. (2 points) Which code fragment could be an example of the concept of data encapsulation?

A. `public E e;`
B. `public E e();`
C. `private E e;`
D. `private E e();`
E. `enclose E e;`

22. (2 points) Which code fragment could be an example of a constructor definition?

A. `R();`
B. `void R() { ... }`
C. `static R() { ... }`
D. `public R() { ... }`
E. `R R() { ... }`

23. (2 points) Which code fragment could be an example of an accessor definition?

A. `setS(S s) { ... }`
B. `S getS() { ... }`
C. `List<S> = new ArrayList<S>();`
D. `t.setS(s);`
E. `t = s.getS();`

24. (2 points) Which code fragment could be an example of a mutator definition?

A. `setS(S s) { ... }`
B. `S getS() { ... }`
C. `List<S> = new ArrayList<S>();`
D. `t.setS(s);`
E. `t = s.getS();`

25. (2 points) Which code fragment could be an example of Generics?

A. `setS(S s) { ... }`
B. `S getS() { ... }`
C. `List<S> = new ArrayList<S>();`
D. `t.setS(s);`
E. `t = s.getS();`

26. (2 points) Which code fragment could be an example of the Java Collections Framework?

A. `setS(S s) { ... }`
B. `S getS() { ... }`
C. `List<S> = new ArrayList<S>();`
D. `t.setS(s);`
E. `t = s.getS();`

27. (2 points) Which code fragment could be an example of subclass assignment?

A. `a.toString();`
B. `objectOutputStream.writeObject(o);`
C. `List<S> = new ArrayList<S>();`
D. `class G extends F { ... }`
E. `super(param1, param2);`

28. (2 points) Which code fragment could show an example of dynamic binding?

A. `a.toString();`
B. `objectOutputStream.writeObject(o);`
C. `List<S> = new ArrayList<S>();`
D. `class G extends F { ... }`
E. `super(param1, param2);`

29. (2 points) Which code fragment could show an example of constructor chaining?

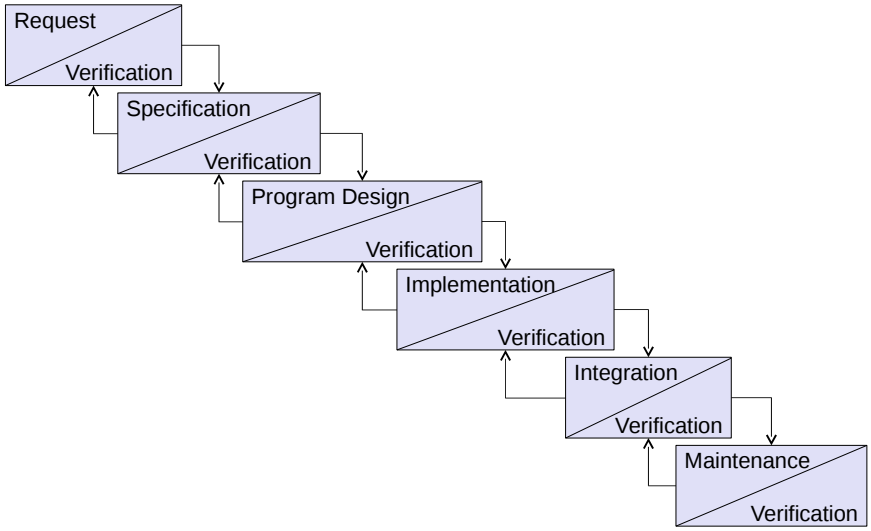
A. `a.toString();`
B. `objectOutputStream.writeObject(o);`
C. `List<S> = new ArrayList<S>();`
D. `class G extends F { ... }`
E. `super(param1, param2);`

30. (2 points) Which code fragment could show an example of serialization?

A. `a.toString();`
B. `objectOutputStream.writeObject(o);`
C. `List<S> = new ArrayList<S>();`
D. `class G extends F { ... }`
E. `super(param1, param2);`

Part III. Abstraction and Program Development

Refer to the following version of the waterfall model of the software lifecycle when answering the questions in this part.



31. (2 points) Which step involves the creation of UML class diagrams?
    A. Request
    B. Specification
    C. Program Design
    D. Implementation
    E. Integration

32. (2 points) Which step involves adding code to the bodies to methods?
    A. Request
    B. Specification
    C. Program Design
    D. Implementation
    E. Integration

33. (2 points) Which step involves putting numerous units together into a larger software system?
    A. Request
    B. Specification
    C. Program Design
    D. Implementation
    E. Integration

34. (2 points) Which step involves formally describing the requirements of the customer?
    A. Request
    B. Specification
    C. Program Design
    D. Implementation
    E. Integration

35. (2 points) Which step involves discovering the desires/needs of the customer?

A. Request

B. Specification

C. Program Design

D. Implementation

E. Integration

Part IV. Coding Conventions

36. (2 points) Which naming choice is most appropriate for a class for which each instance is intended to represent an individual item?

A. A singular noun

B. A plural or collective noun

C. A verb

D. An adjective

E. An adverb

37. (2 points) Which naming choice is most appropriate for a class for which each instance is intended to represent a group of items?

A. A singular noun

B. A plural or collective noun

C. A verb

D. An adjective

E. An adverb

38. (2 points) Which naming choice is most appropriate for a method?

A. A singular noun

B. A plural or collective noun

C. A verb

D. An adjective

E. An adverb

39. (2 points) Which naming choice is most appropriate for an interface that describes a characteristic of implementing classes?

A. A singular noun

B. A plural or collective noun

C. A verb

D. An adjective

E. An adverb

40. (2 points) Which naming choice is most appropriate for a field that holds data for an individual item?

A. A singular noun

B. A plural or collective noun

C. A verb

D. An adjective

E. An adverb

Part V. The Java Collections Framework

41. (2 points) What is one difference between the `ArrayList` and `Vector` classes?
    A. `ArrayList` requires a contiguous block of memory; `Vector` does not.
    B. The elements of `Vector` must be unique; `ArrayList` allows for duplicates.
    C. `Vector` can be used concurrently without data corruption risks; `ArrayList` cannot.
    D. `ArrayList` is older than `Vector` and is gradually being replaced by `Vector`.
    E. `Vector` takes both a key and value; `ArrayList` just takes elements.

42. (2 points) What is one difference between the `ArrayList` and `LinkedList` classes?
    A. `ArrayList` requires a contiguous block of memory; `LinkedList` does not.
    B. The elements of `LinkedList` must be unique; `ArrayList` allows for duplicates.
    C. `LinkedList` can be used concurrently without data corruption risks; `ArrayList` cannot.
    D. `ArrayList` is older than `LinkedList` and is gradually being replaced by `LinkedList`.
    E. `LinkedList` takes both a key and value; `ArrayList` just takes elements.

43. (2 points) What is one difference between the `ArrayList` and `TreeSet` classes?
    A. `ArrayList` requires a contiguous block of memory; `TreeSet` does not.
    B. The elements of `TreeSet` must be unique; `ArrayList` allows for duplicates.
    C. `TreeSet` can be used concurrently without data corruption risks; `ArrayList` cannot.
    D. `ArrayList` is older than `TreeSet` and is gradually being replaced by `TreeSet`.
    E. `TreeSet` takes both a key and value; `ArrayList` just takes elements.

44. (2 points) What is one difference between the `ArrayList` and `TreeMap` classes?
    A. `ArrayList` requires a contiguous block of memory; `TreeMap` does not.
    B. The elements of `TreeMap` must be unique; `ArrayList` allows for duplicates.
    C. `TreeMap` can be used concurrently without data corruption risks; `ArrayList` cannot.
    D. `ArrayList` is older than `TreeMap` and is gradually being replaced by `TreeMap`.
    E. `TreeMap` takes both a key and value; `ArrayList` just takes elements.

45. (2 points) Why are both sets and maps implemented using either hash tables or trees?
    A. Both the elements of sets and the keys of maps must be unique.
    B. Hash tables and trees both use a similar underlying array.
    C. The values of maps are stored in sets.
    D. Both are part of the Java Collections Framework.
    E. Both sets and maps are subclasses of `List`.

Part VI. Searching and Sorting

46. (2 points) Which is more efficient for large lists?
    A. Copying a `LinkedList` to an `ArrayList`
    B. Copying an `ArrayList` to a `LinkedList`
    C. Performing a binary search on an already sorted `ArrayList`
    D. Performing a linear search on a `LinkedList` that may be sorted or unsorted
    E. Sorting an `ArrayList` and then performing a binary search

47. (2 points) Which of the following statements is true?
    A. In order to successfully perform a binary search on a list, the list must be sorted.
    B. A linear search may be performed on a linked list or an array.
    C. A stable sort will preserve a list's prior order, except as necessary to satisfy the sort criteria.
    D. All of the above.
    E. None of the above.

48. (2 points) Which of the following statements is true?
    A. A `Comparable` class has a natural ordering defined by its `compareTo` method.
    B. Classes may implement `Comparator` to provide additional methods for comparing objects.
    C. A `TreeSet` is always maintained in sorted order.
    D. All of the above.
    E. None of the above.

49. (2 points) Which of the following statements is true?
    A. A `compare` method should return `true` if the two objects are the same
    B. A `compare` method should return `false` if the two objects are different
    C. To sort a list, the objects in the list must implement `Comparable`
    D. All of the above
    E. None of the above

50. (2 points) The most likely outcome of doing a binary search for an item that is in an unsorted list is which of the following?
    A. Quickly finding the item
    B. Slowly finding the item
    C. Not finding the item
    D. Finding the wrong item
    E. Throwing `IndexOutOfBoundsException`