

Unit 5 + 6 Assessment will cover everything up to this point, but the focus is on the last topics: All the OOP concepts, Exceptions, Collections, Maps. Functional Interface and Lambda Expression will **NOT** be on this assessment.

Time Control: 1 hour and 30 minutes (90 minutes) | Once you begin, the time remaining will be shown on the right side under the questions
Questions: Just under 30 questions

Available: Saturday, June 4th @ 5 pm (after class)
Due: Saturday, June 11th by 11:59 pm EST

To find the assessment:

Go to Canvas → Java programming course → Quizzes → Unit 5 + 6 Assessment

Note: Assume all classes used are imported.

No code will fail to compile because of missing import or main method. Focus only on the given code snippet. Assume everything else not shown is valid.

[5.1] Encapsulation

- What is Encapsulation?
 - Access modifiers in relation to variables, methods, classes
 - Immutability (How can it be achieved)
 - Hiding data, providing public access
-

[5.2] Inheritance

- What is Inheritance?
 - Relation of the super/sub class – aka parent/child
 - Rules for super and sub classes
 - Which properties are passed on and what is the influence of access moodier on those properties?
 - Static members in inheritance
 - How does protected access modifier behave regularly and alongside inheritance?
-

[5.3] Overriding

- What is Overriding
 - Overriding vs Overloading
 - Rules of overriding
 - Method calling in relation to overriding implementation
-

[5.4] Exceptions

- What is the purpose of Exceptions?
 - Checked vs Unchecked Exceptions
 - High level idea of the inheritance of Exceptions
 - Try/catch block
 - Finally keyword
-

[5.5] Final

- Final keyword
 - Rules for final method, variables, classes
 - Difference between final, finally, finalize
-

[6.1] Abstraction

- What is Abstraction
 - Properties of an abstract class
 - Properties of interface
 - Properties of an abstract method
 - Abstract class vs interface comparison
 - Abstraction in relation with concrete classes and other abstract classes
 - Interface inheritance
-

[6.2] Polymorphism

- What is Polymorphism
 - Reference type vs object type
 - Possible reference types for Objects
 - Casting (up and down)
-

[6.3] Collections

- What is the collection framework?
 - Main concept of the interfaces: List, Set, Queue
 - Create classes that implement the interfaces
 - What is synchronized/ thread safe
-

[6.4] Maps

- Main concept of the Map interface
 - Create classes that implement the interface
 - Map methods
 - Looping through a map
-