Java programming optional assignment

This assignment will act as a substitute for those who missed class session assignments.

Instruction: ***Answer all questions***

1. What are the different logical operators available in Java? Provide examples of how they are used.
2. How does the `&&` (logical AND) operator differ from the `&` (bitwise AND) operator in Java? Write a small program to demonstrate the difference.
3. Explain the short-circuit behavior of the `&&` and `||` operators in Java. How does it impact the performance of conditional statements? Provide a code example.
4. What is the difference between the `equals()` method and the == operator in Java? Write a program to compare two objects using both.
5. How should the `equals()` method be overridden in a custom class to ensure proper comparison of objects? Write a Java class that demonstrates this.
6. Why is it important to override the `hashCode()` method when overriding the `equals()` method? Provide a practical example to illustrate the importance.
7. What are some of the key features of Java that make it a widely-used programming language?
8. How does Java achieve platform independence? What role does the Java Virtual Machine (JVM) play in this? Write a short program and explain how it runs on different platforms.
9. Explain the concept of garbage collection in Java. How does it help in memory management? Write a program that triggers garbage collection.
10. What is the difference between static and non-static methods in Java? Provide examples in a Java class.
11. Can a static method access instance variables in Java? Why or why not? Write a program to demonstrate this.
12. Write the syntax for creating a static method and a non-static method in Java. Create a class with both types of methods and explain their differences.
13. How do instance variables differ from arrays in Java? Provide an example to illustrate the differences.
14. Can arrays in Java hold different data types? Explain with an example program.
15. What are the advantages of using arrays over individual instance variables? Write a program that demonstrates these advantages.
16. What is a generic class in Java? Provide an example of how to define and use one.
17. How do generic methods differ from generic classes in Java? Provide a use case and write a program to demonstrate it.
18. Why is it beneficial to use generics in Java? Explain with an example program that shows the advantages of using generics.
19. What is static binding in Java? When does it occur? Provide an example to illustrate static binding.
20. Explain dynamic binding in Java with an example. How is it different from static binding? Write a program to demonstrate dynamic binding.
21. How does Java determine whether to use static or dynamic binding for a method call? Provide an example to clarify.
22. How do you open a file for reading using the `BufferedReader` class in Java? Write a program to read a file line by line.

23. Write a Java code snippet to read a file line by line using `BufferedReader`. Handle any possible exceptions that might occur.
24. Explain how to handle exceptions when working with file I/O in Java. Write a program that reads a file and handles exceptions appropriately.
25. What is method overloading in Java? How does it differ from method overriding? Write a class that demonstrates both concepts.
26. Provide an example of method overloading in Java. Write a program with multiple overloaded methods and explain their use cases.
27. What are the rules for method overriding in Java? Provide an example by creating a superclass and a subclass that overrides a method.
28. What is the difference between a process and a thread in Java? Provide an example program to create and start a thread.
29. How do you create and start a thread in Java? Provide a code example that demonstrates thread creation and starting.
30. Explain the concept of thread synchronization in Java. Why is it important? Write a program that uses synchronized methods to manage thread access to shared resources.
31. What is an `ArrayList` in Java? How does it differ from an array? Write a program that demonstrates the use of an `ArrayList`.
32. How do you create an `ArrayList` in Java? Provide an example program that initializes an `ArrayList` and adds elements to it.
33. Write a Java code snippet to add items to an `ArrayList`. Demonstrate adding different types of elements to the list.
34. What is the difference between an abstract class and an interface in Java? Provide examples of each and explain their use cases.
35. Can an abstract class have a constructor? Explain why or why not, and provide a program to demonstrate your explanation.
36. How do you implement an interface in a Java class? Provide an example program that implements multiple interfaces.
37. Explain the concept of inheritance in object-oriented programming with an example in Java. Write a program that demonstrates class inheritance.
38. What is the "has-a" relationship in Java? How does it differ from the "is-a" relationship? Provide an example to illustrate both.
39. How does polymorphism support object relationships in Java? Provide an example program that demonstrates polymorphism.
40. What are checked and unchecked exceptions in Java? Provide examples of each and explain how they differ.
41. How does Java enforce exception handling for checked exceptions? Write a program that demonstrates handling a checked exception.
42. Explain how to create a custom checked exception in Java. Write a program that defines and uses a custom exception.
43. What is an inner class in Java? How does it differ from a regular class? Provide an example program that uses an inner class.
44. How do you define an interface in Java? Provide an example program that defines and implements an interface.
45. Can an interface have inner classes? Explain with an example program.
46. What is modularity in Java? How does it benefit large-scale applications? Write a simple Java module and explain its components.
47. How do you create a module in Java? Provide an example program that includes multiple modules and explains how they interact.

48. Explain the role of the `module-info.java` file in Java modularity. Write an example module with a `module-info.java` file and explain its content.
49. How do you sort an array of objects in Java? Provide an example using the `Comparable` interface to sort objects.
50. Write a Java code snippet to delete an item from an `ArrayList`. Demonstrate removing elements by index and by value.
51. How do you pass an object as an argument to a method in Java? Provide an example program that demonstrates passing and modifying an object in a method.