1.

a) Checked exceptions happen at compile time when the source code is transformed into an executable code. Unchecked exceptions happen at runtime when the executable program starts running.

b)

The 'super' keyword in Java enables referencing the superclass or parent class of a subclass. It's commonly used to access members such as fields or methods of the superclass that have been overridden in the subclass. By using 'super', you can invoke the superclass's method from within the subclass.

c)

Generics in Java help make types more flexible. They let classes, interfaces, and methods work with different types of objects. Generics make sure our code is safer, we don't have to keep casting types all the time, and they make it easier to reuse code

2.

a)

SOLID is like a guide for making software that's organized and simple to handle. It helps you keep your code clean, understandable, and easy to fix or change later. Following SOLID can save you time and energy when you're building or updating software. It stops your code from getting too stiff or fragile, so your software stays strong and reliable for a long time.

b)

Lambda expressions are like shortcuts in Java. They let you write small, neat chunks of code that do specific tasks. They came with Java 8 and are great for making your code shorter and easier to understand. Think of them as mini methods that make your code look cleaner and more straightforward.

c)

There are numerous design patterns used in Java, but some of the most important and widely used ones are:

1. Singleton pattern
2. Factory pattern
3. Observer pattern
4. Decorator pattern
5. Adapter pattern

6. Strategy pattern
7. Template method pattern
8. Iterator pattern
9. Composite pattern
10. Proxy pattern

    Each pattern has a specific purpose and can be used to solve common problems in software development. It's important to understand these patterns and their appropriate use to write efficient and maintainable code.