1. Checked exceptions are the kind of exceptions that occur at compile time and are also known as compile time exceptions and should be taken care by the programmer since they cannot be ignored during program compilation. The exceptions must be handled either through a catch and try block or by using the throw clause.

   Unchecked exceptions are the kind of exceptions that occurs during program execution and are ignored during program compilation. Also known as runtime exceptions.

   Examples

   Checked Exception

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;


public class checkedException

{

  public static void main ( String[] input ) throws IOException

  {

    BufferedReader stdin =  new BufferedReader ( new
InputStreamReader( System.in ) );


    String in ;

    in = stdin.readLine();


    // More Java code ...

  }

}
```

   Unchecked Exception

```
package anagramChecker;



public class UncheckedException

{
    public static void main(String[] args)
    {
        try{
          int i = Integer.parseInt("Unchecked Exception");


          //Above statement throws NumberFormatException which is an
unchecked    exception
        }
        catch(NumberFormatException e){
            System.err.println("Numberformat exception caught here");
        }
    }
}
```

3. Integer is a class in java. An integer variable stores a reference to an integer object. In java, two references are equal if they point to the same object in terms of == an are not equal if they pint to different objects in term of ==. In this case, both cases should return false. However, the Integer.java class has an inner private class, IntegerCache.java that caches all Integer objects between -128 and 127. In the first assignment, the Integer object 1000 is not in the range -128 -127 therefore it is not cached. In the second instance of assignment, the Integer object 100 falls in the range -128-127 therefore it is cached. What Integer a=100 does is create an object with value 100 in memory and and a refers to it. When another object, b=100 is created, a new object with value is 100 is created. When performing the == operation, the object referenced by a is retrieved and since it i within the cachable range, it is cached. When retrieving the value of b, it assigned the cached value, 100 and therefore both references will pointing to the same object and therefore a==b returns true.

4. The Model View Controller(MVC) pattern, breaks dowd the different component of the application with the aim of separating responsibilities. The View contains the visual layout and presentation of the application, the Model contains the data, logic and rules of the application and the Controller connects the two.

What MVC does is separate the application logic from the user interface. Requests are passed to the controller then the controller works with the model to work on the data then passes back to the view. The view then displays the data to the user.

5. Arraylist and linkedlist are both implementations of List Interface of the collection framework in java. They can be used interchangeably since they both function the same. However, they also have some differences that makes each of them unique and one suitable for certain operations over the other.Accessing elements is faster in an arraylist than a linked list because it is indexed based whereas a linked list is not. On the other hand insertion and deletion in a linked list is faster than arraylist. All you need is the node of the linked list then changes the pointers just before or after the node. For an arraylist, the indexes according to the deletion or insertion if the operation is on middle indexes. This differences s what makes each of the two to be suitable or unsuitable for a particular situation over the other. Generally, an arraylist is ideal for a situation where data is accessed more frequently than deletion and insertion and vice versa for a linkedlist. Following are real world examples on when to use which.

When to use a linked list

a. A queuing system where insertion is done in one end and deletion is done on the other end. For instance in and the operations happens frequently for instance in a mess queue.
b. Memory management in java and c languages applies a linked list where a node represents available or used block of memory.
c. A SMS or emailing sending system will use a linked list to queue the SMSs or emails.
d. A Medical imaging application will use a link list to queue images that has to be burned to a CD.

e. A multiple player board game uses a linked list to manage the players to access game objects.
f. The OS uses a linked list to manage time sharing processes.

When to use an arraylist

6. Benefits of using Spring

a. Lightweight
b. Inversion of control for loose coupling
c. Has generic abstraction layer for Transaction management
d. JDBC abstraction layer which simplifies JDBC Exception handling
e. Support for MVC framework
f. Support for aspect oriented programming

Spring Subprojects I know

a. Spring boot.
b. Spring security.
c. Spring Integration.

8. Dependency injection is a technique in programming where one object provides the dependencies of another object. Dependency injection ensures that no changes can be made on client when the object that the client depends on has been changed. The client delegates the responsibility of the work of providing its dependencies to an external object (injector). The injector calls the client to inject the service. The client doesn't know how to construct the services. Neither does it know which service it is using. All it has to know is the interfaces of the service since this is what defines how the client may use the service. This way, dependency injection separates the responsibility of construction and use.

10. Benefits of Hibernate Framework

a. Persistent transparency- This ensures automatic connection between the objects of the application with the database tables enabling reduction in development time and maintenance cost.
b. Support for Hibernate query language which is object oriented with advanced features such as pagination and dynamic profiling.
c. Lazy loading- allows fetching of the objects that are required by the application enhancing its performance.
d. Version property- Hibernate supports optimistic locking by use of the version property which allows multiple transactions to be executed without affecting each other.
e. Dual-layer caching- Supports both single and double layer caching. This reduces the number of hits to the database making hibernate highly scalable and increases the applications performance.