Interface and Lambda expression.

(1.) what is an interface in java?

Ans:- An interface in java is a mechanism that is used to achieve complete abstraction. It is basically a kind of class that contains only constants and abstract methods.

(2) which modifier are allowe for methods in an interface? Explain with example.

Ans:- only abstract and public modifier are allowed for methods in interface.

(3) what is the use of interface in java?

Ans:- There are many reasons to use interface in java. They are as follows:

(a) An interaface is used to achieve full abstraction.
(b) using interface is the best way to express our project's API to some other project.
(c) programers use interfaces to customizes features of software differently for different objects.
(d) By using interface, we can achieve the functionality of multiple interface.

(4) what is the difference between abstract class and interface in Java?

(8) Example

```
public abstract class shape
  public abstract void draw();
}
```

(8) Example.

```
public interface Drawable
      void draw();
}
```

(1) what is lambda expression of Java 8?

Ans:- As its name suggests it's an expression which allows you to write more succinct code in Java 8. For example: (a,b) -> a+b is a lambda expression (look for that arrow ->).

which is equal to following code.

```
public int value (int a, int b) {
      return a+b;
}
```

It is also called an annonymous function because you are essentially writing the code you write in' function but without name.

(2) can you pass expression to a method? when?

Ans:- Yes. you can pass a lambda expression to a method provided it is expecting a functional interface. For example. If a method is accepting a Runnable, comparable or comparator then you can pass a lambda expression to it because all these are functional interfaces in Java 8.

(3) what is the functional interface in Java 8?

Ans:- A functional interface in Java 8 is an interface with a single abstract method. For example, comparator which has just one abstract method called compare () or Runnable which has just one abstract method called run(). There are many more general purpose functional interface introduced in JDK on java. util. function package.

They are also annotated with @ Functional interface but that's optional.

(4) what is the benefits of lambda expressions in Java8?

Ans:- The main benefits of lambda expression in Java8 is that now it's easier to pass a code block to a method. Earlier, the only way to do this was wrapping the code inside an Anonymous class, which requires a lot of boilerplate code.

(5) Is it manadatory for a lambada expression to have parameters?

Ans → No, it's not manadatory for a lambda expression to have parameters. you can define a lambda expression without parameters as shown below:

() → syso("lambda without parameter");

you can pass this code at any method which accepts a functional interface.

Abstract class	Interface
① Abstract class can have abstract and nonabstract method.	① Interface can have only abstract methods. Since Java 8 it can have static and default method also.
② Abstract class doesn't support multiple inheritence	② Interface support multiple inheritence.
③ Abstract class can have final, non-final, static and non-static variables.	③ Interface has only static and final variables
④ Abstract class can provide the implementation of inheritence	④ Interface can't provide the implementation of abstract class.
⑤ The abstract keyword is used to declair abstract class	⑤ The interface keyword is used to declair the interface
⑥ An abstract class can extends anothe Java class and implement multiple Java interfaces	⑥ An interface can extends another Java interface only.
⑦ An abstract class extended using the keyword extends	⑦ An interface can be implemented using the implements.
⑧ A Java abstract class can have class members like private, protected etc.	⑧ members of a Java interface are public by default.