Text

Description automatically generated

A: D, generics promote all of these values because they are type-safe meaning they can take any datatype, they’re reliable because they can handle legacy code and they are reliable being able to handle an array of data types. That’s why the answer is D because it has all of these characteristics.

A picture containing text

Description automatically generated

A: No because a generic will only work with a reference type, not the primitive type.



class FlightSched<T,V> { // function body}



class FlightSched<T,V extends Thread> { // function body}



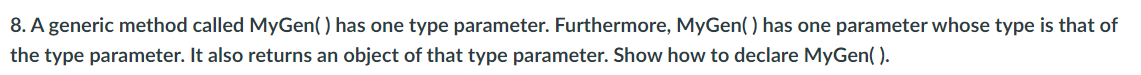
class FlightSched<T,V extends T> { // function body}



The “?” operator is referred to as the **wild card** operator. This helps bound certain data types to have an upper or lower bound.



Yes, as I mentioned in 6, a wildcard can have an upper or lower bound.



<T> T MyGen(T o) {// Function body}

Graphical user interface, application

Description automatically generated

class MyClass<T, V extends T> implements IGenIF<T, V> { // Function body }

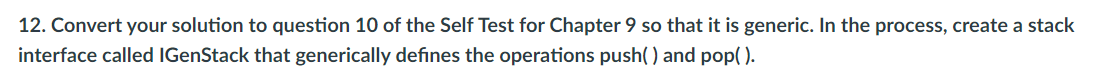


Counter obj = new Counter;

A picture containing text

Description automatically generated

A: No, a process called erasure takes place which is the instance in which all type parameters are erased during compilation because the actual objects will be passed during run-time.



Interface

A screenshot of a computer

Description automatically generated with medium confidence

StackFullException class

Text

Description automatically generated

StackEmptyException class

Graphical user interface

Description automatically generated with medium confidence

GenStack class

Text

Description automatically generated

Text

Description automatically generated

13.



The diamond operator which is used to for parameterization purposes with generics.

14.

Graphical user interface, website

Description automatically generated

By typing the following

MyClass<Double, String> obj = new MyClass<>(1.1, “Hi”);