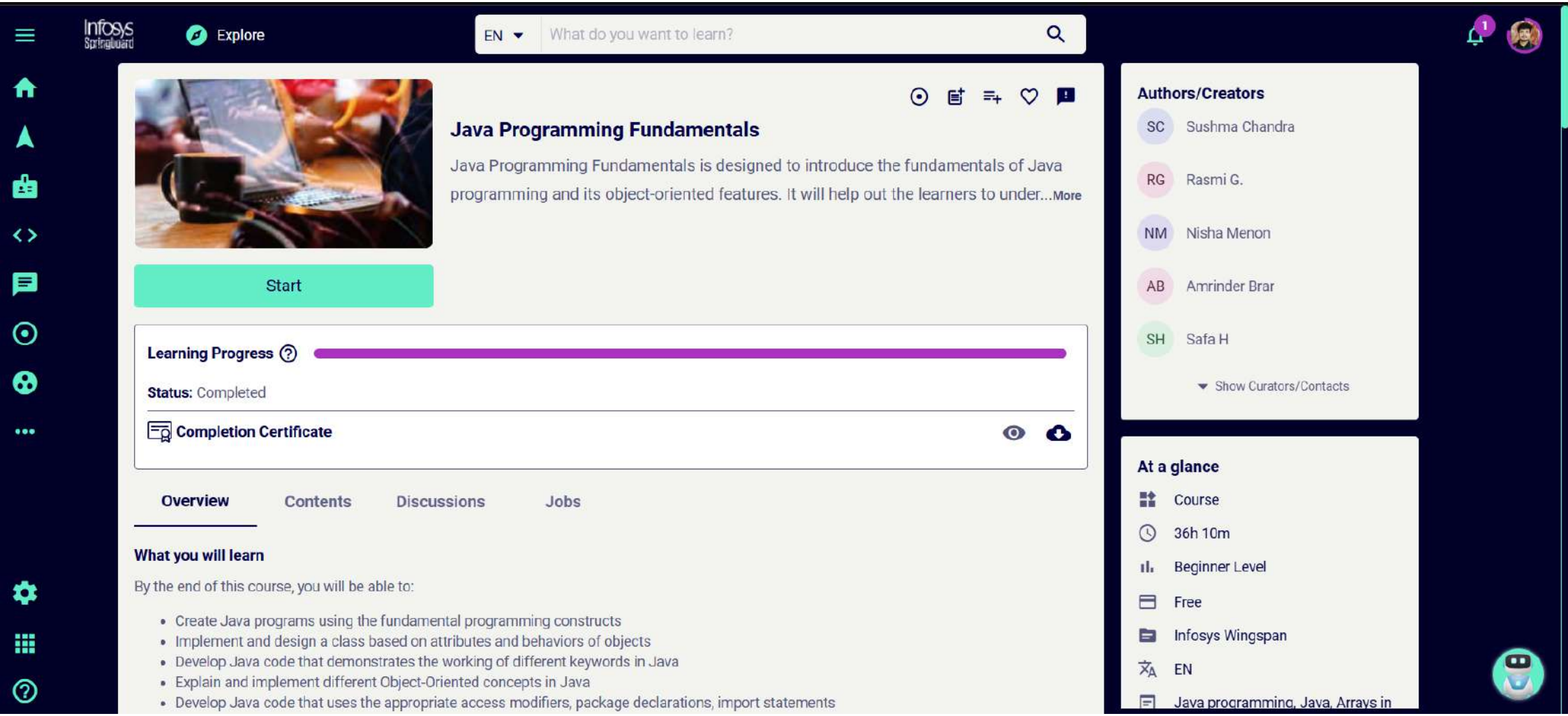


Assessment - Java Programming Fundamentals

Thank you. Your test submitted.

You have cleared this assessment.

Obtained Percentage	Obtained Marks
72.55 %	37 / 51

Best Attempt Score:72.55 % on 17-02-2026

Review Your Attempt

What is the output of the below code?

```
abstract class BaseAbsClass {
	public void method() {
		System.out.println("BaseAbsClass Method");
	}
	public abstract void method2();
}

class Derived extends BaseAbsClass {
	public void method2() {
		System.out.println("Derived met
	}
}

public class Testing {
	public static void main(String[] args
		BaseAbsClass obj = new Derived(
		obj.method2();
	}
}
```

- (●) Derived method
- ( ) Compilation Error: Cannot have non abstract method in an abstract class
- ( ) Compilation Error: Cannot have a reference to Abstract class
- ( ) BaseAbsClass Method

What is the output of the following code?

```
abstract class Person1 {
    public final void display(){
        System.out.println("Display method in Person");
    }

    public static void main(String[] args){
        Person1 person = new Student1();
        person.display();
       }
}

public class Student1 extends Person1{
      public void display(){
           System.out.println("Display met
      }
}
```

Warning

This operati

- ◯ Display method in Person
- ◯ Display method in Student
- ◯ Code will not compile because of line 7
- ◯ Code will not compile because of line 11

What is the output of the following code?

```
class Person{
    public Person(String name){
          System.out.println(name);
  }
}

public class Student extends Person{
       public Student(){          //line 8
             System.out.println("Student");
       }
       public static void main(String[] args
            new Person("Bob");
       }
}
```

- BobStudent
- Bob
- The code will not compile due to line 8
- The code will not compile due to line 11

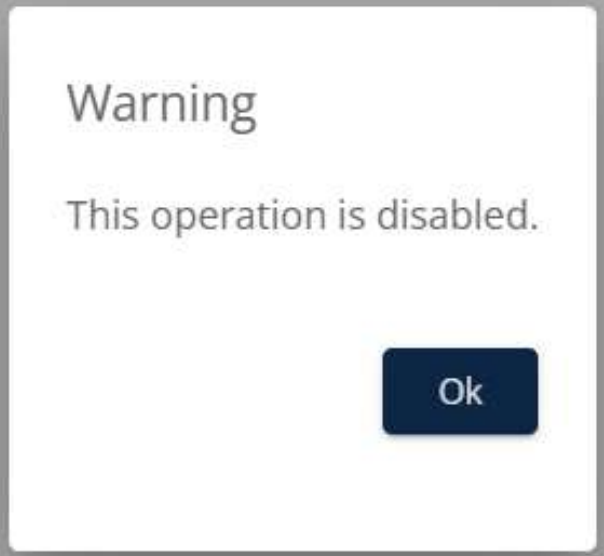
What will be the output of the below code?

```
public class Test {
        public static void main(String[] args) {
                try {
                        System.out.print("In try ");
                        return;
                }
                finally {
                        System.out.print("In finally ");
                }
                System.out.print("Outside block ");
        }
}
```

- ○ Compilation error
- ◉ In try In finally
- ○ In try In finally Outside block
- ○ Runtime error

Predict the output of the below code:

```
package generic;
import java.util.ArrayList;
import java.util.List;

public class MyListGeneric<Object> {
    private List<Object> values=new ArrayList<>();

    public void add(Object value) {
        values.add(value);    //line 1
        System.out.println(values);
    }

    public static void main(String[] args) {
                MyListGeneric<String> myListString = new
                myListString.add("Good");
        }
}
```



- ◯ The add method accepts only Object type, hence make the myListString to accept only Object types
- ◯ The code will lead to compilation error at line 1
- ◯ String is immutable and hence explicit String declaration should have been done in the class declaration as MyListGeneric<String>
- ◉ [Good]

Which of the following is true with respect to StringBuilder in Java?

(Choose 2 options)

- [x] StringBuilder is non-synchronized, not thread safe
- [ ] StringBuilder is synchronized, thread safe
- [ ] StringBuilder is less efficient
- [x] StringBuilder is more efficient

Warning

This operation is di

Which of the following statements are correct advantages of an immutable class?

a. Immutable objects are simple
b. Immutable classes are thread-safe, they require no synchronization

- ○ only a is correct
- ○ Only b is correct
- ◉ Both a and b are correct
- ○ Both a and b are incorrect

Warning

This operation is di

What is the output of the below code?

```
public class Tester {
	public static void main(String[] args) {
		Base obj = new Derived();
		obj.method(25);
	}
}

class Base {
	public static void method(int a) {
		System.out.println("Base Method
	}
}

class Derived extends Base {
	public static void method(int a) {
		System.out.println("Derived Met
	}
}
```

Warning

This operatic

- (●) Base Method
- ( ) Derived Method
- ( ) Compilation Error
- ( ) Runtime Exception

Given:

```
package p1;
class A { }

package p2;
import p1.A;
class B extends A { }
```

What is the output?

- (●) class A & class B compiles
- ( ) Both class A and B compile but class B shows some w
- ( ) Compilation of class B fails
- ( ) class B throws an exception

Which of the following is the correct way of returning an empty collection?

1.
```
List<String> list;
public List<String> getlist() {
        if (list.size() == 0)
                return null;
        else
                return list;
}
```

2.
```
List<String> list;
public List<String> getlist() {
        if (list.size() == 0)
                return Collections.emptyList();
        else
                return list;
}
```

Warning

This operation

- ○ Only a is correct
- ◉ Only b is correct
- ○ Both a and b are correct
- ○ Both a and b are incorrect

Which of the following lines of code compile?

(Choose any 4 options)

- [x] int number = 1234;
- [ ] float f1 = 1234.0;
- [x] float f2 = 1234;
- [x] double d3 = 1234.0;
- [x] long num = 1234;

What is the output of the below code?

```
interface MyInterface {
	void method1();
}

class MyImplementation implements MyInterface {
	void method1() {
		System.out.println("My Method")
	}
}

public class Testing1 {
	public static void main(String[] args
		MyInterface obj = new MyImpleme
		obj.method1();
	}
}
```

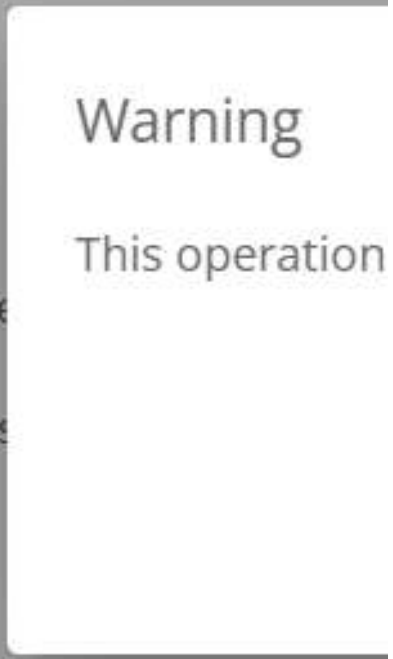
Warning

This operati

- ○ My Method
- ◉ Compilation Error: cannot reduce the visibility of inherited method
- ○ Compilation Error: cannot have non-public methods in MyInterface
- ○ Compilation Error: cannot have non-public methods in MyInterface

What is the output of the below code?

```
class Vehicle {
	Vehicle() {
		System.out.println("Vehicle is created");
	}
}

public class Bike5 extends Vehicle {
	Bike5() {
		super();
		System.out.println("Bike is cre
	}
	public static void main(String[] args
		Bike5 b = new Bike5();
	}
}
```



- (●) Vehicle is created Bike is created
- ( ) Vehicle is created Vehicle is created Bike is created
- ( ) Bike is created Vehicle is created
- ( ) Vehicle is created Bike is created Vehicle is created

```
import java.util.*;

public class App {
        public static void main(String[] args) {
                Set<String> s = new TreeSet<String>();
                s.add(2);
                s.add("1");
                s.add("3");
                Iterator i = s.iterator();
                while (i.hasNext()) {
                        System.out.println(i.next() + " "
                }
        }
}
```

Warning

This operation is di

What will be the output?

- 1 2 3
- 2 1 3
- ClassCastException: java.lang.String cannot be cast to java.lang.Integer
- Compile time error

If a class's methods and variables are to be made accessible only to its sub classes in different packages, what is the suitable access specifier?

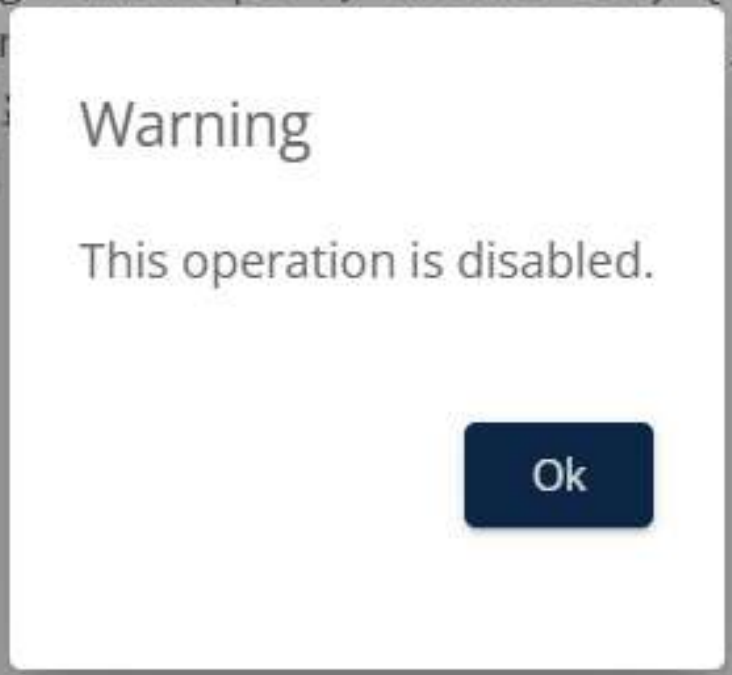
- (●) protected
- ( ) default
- ( ) public
- ( ) private

Warning

This operation is disabled

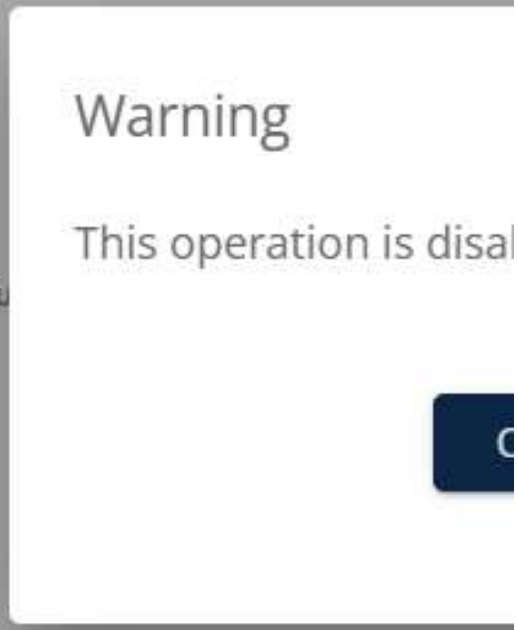
Predict the output of the following code

```
public class VarargsTest {
	public static void main(String[] args) {
		displayRegn("Hockey"); // Line 1
		displayRegn("Kho-Kho", 1, 2, 3);
	}

	public static void displayRegn(String nameOfSport, int... iDs) {
		System.out.print(" Registratio
		for (int i = 0; i < iDs.length;
			System.out.print(iDs[i]
		}
	}
}
```



- ○ Registration for Hockey: Registration for Kho-Kho: 1 2 3
- ◉ Registration for Kho-Kho: 1 2 3
- ○ Compilation Error, missing parameters
- ○ Runtime Exception

```
import java.util.*;

public class MapDemoCheck {
        public static void main(String[] args) {
                Map<Integer, String> map = new TreeMap<Integer, String>();
                map.put(2, "A");
                map.put(1, "B");
                map.put(3, "C");
                map.put(null,"E");  //1
                map.put(4,null);    //2
                map.put(1,"D");     //3
                Collection <String> collection = map.valu
                for(String element : collection){
                        System.out.println(element);
                }
        }
}
```



What is the output?

- Compile time error, TreeMap cannot contain null as a value as in line 2
- Compile time error, TreeMap cannot have duplicate value for a key as in line 3
- Run time error, TreeMap cannot contain null as key as in line 1
- Will run successfully and Map will contain 4 elements

What is the output of the below code snippet?

```
import java.time.LocalDate;

public class DateDemo {

    public static void main(String[] args) {
        LocalDate date = LocalDate.parse("20
        LocalDate date1 = LocalDate.of(2019,
        System.out.println(date.getYear()+da
        +" , "+date.compareTo(date1)+" , "+c
    }
}
```

Warning

This operatic

- (●) 4038,-9,9
- ( ) Compilation Error
- ( ) 4038,9,-9
- ( ) 4038,0,0

What is the output of the following code?

```
public class SuperClass {
	private void displayName() {
		System.out.println("Super Class");
	}
	public static void main(String[] args) {
		SuperClass superClass = new SubClass();
		superClass.displayName();
	}
}

public class SubClass extends SuperClass {
	private void displayName() {
		System.out.println("SubClass is
	}
}
```

Warning

This opera

- ◯ SubClass is a type of SuperClass
- ◯ Super Class
- ◉ Compilation Error
- ◯ No Error, but no output also

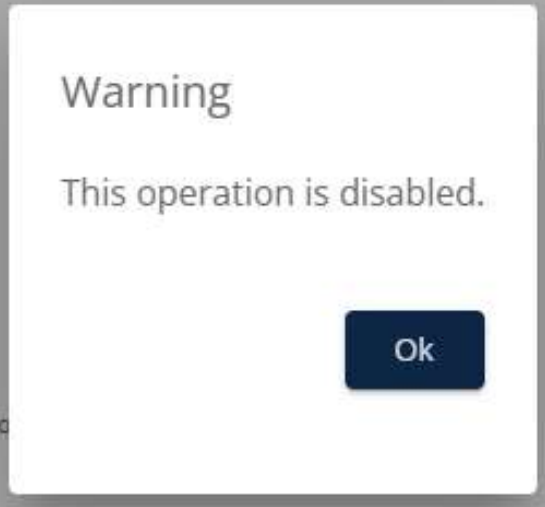
```
import java.util.*;

public class SetImpl {
        public static void main(String[] args) {
                Set<String> set=new TreeSet<String>();
                set.add("Infosys");
                set.add("Google");
                set.add("IBM");
            for(String s:set){
                 System.out.print(" "+s);
                 set.clear();
            }
        }
}
```

What is the output?

- ○ IBM
- ○ Runtime Exception
- ◉ No output
- ○ Google

Which among the following is true with respect to the below code?

```
package generic;
import java.util.ArrayList;
import java.util.List;

public class MyListGeneric<T extends Number> {
    private List<T> values = new ArrayList<>();

    public void add(T value) {
        values.add(value);
        System.out.println(values);
    }
    public void remove(T value) {
        values.remove(value);
    }
    public T get(int index) {
        return values.get(index);
    }

        public static void main(String[] args) {
                MyListGeneric<Float> myList = new MyListGeneric<Flo
                myList.add(98.7f);
                myList.add(1009);
        }
}
```



- (●) The code will lead to a compilation error in the line "myList.add(1009);"
- ( ) The code "myList.add(1009);", if replaced with "myList.add(new Integer(1009));", works fine without compilation error
- ( ) Integer types are subtypes of Number type specified in the MyListGeneric class implementation, hence the program works fine without any errors
- ( ) The int value 1009 will be internally converted to float and will be added to myList, hence the program works fine without any errors

Which is the correct way of comparing the contents of the below strings?

```
String firstString = "String";
StringBuilder secondString = new StringBuilder("String");
```

- ○ firstString == secondString
- ○ firstString.contains(secondString)
- ◉ secondString.equals(firstString)
- ○ firstString.equals(secondString)

Warning

This oper

What is the result when the following code is compiled?

```
public class Test {
    public void method() {
        for(int i = 0; i < 3; i++) {
            System.out.print(i);
        }
        System.out.print(i);
    }
}
```

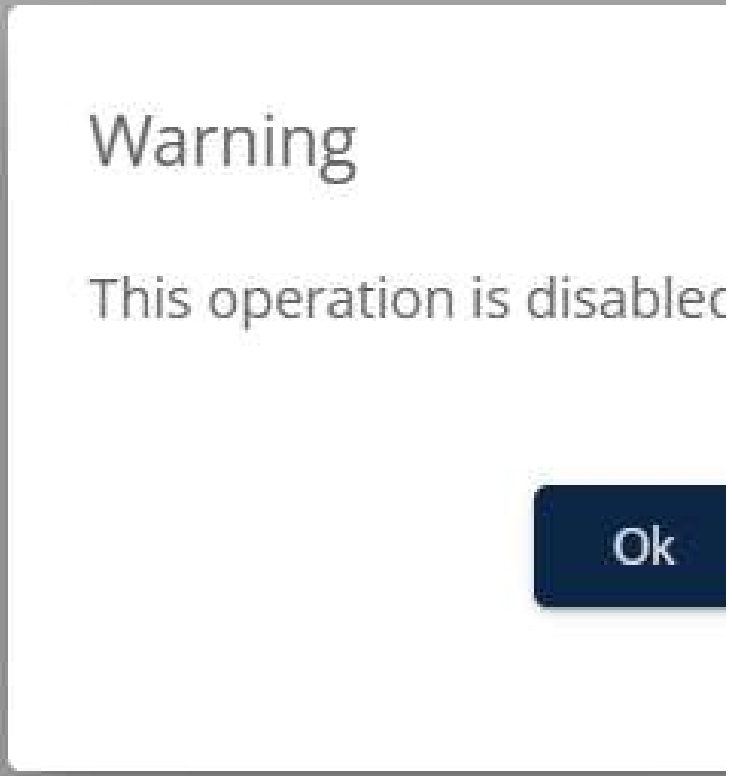
Warı

This c

- 0123
- 123
- Compilation error
- Runtime error

How many objects are eligible for garbage collection in the below code?

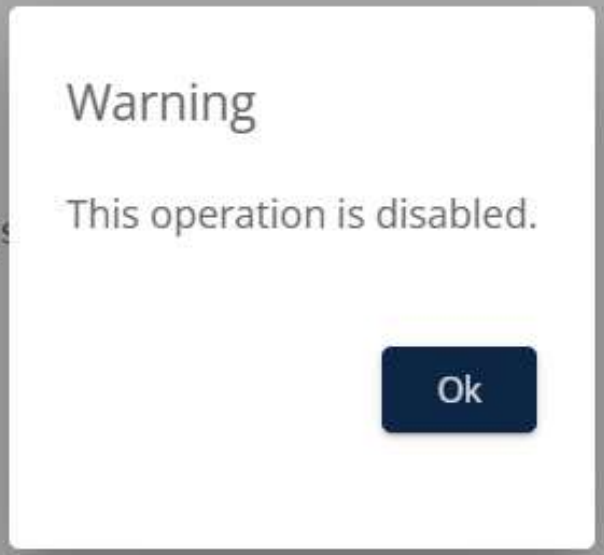
```
public class Loan {
	public static void main(String[] arg) {
		Loan loan1 = new Loan();
		Loan loan2 = new Loan();
		Loan loan3 = new Loan();
		Loan loan4 = new Loan();
		loan1=loan3;
		loan2=loan4;
		loan3=null;
	}
}
```



- ○ 0
- ○ 1
- ◉ 2
- ○ 3

Given the enum and the Java class

```
public enum Day {
	SUNDAY(1), MONDAY(2), TUESDAY(3), WEDNESDAY(4), THURSDAY(5), FRIDAY(6), SATURDAY(7);
	private int value;
	private Day(int value) {
		this.value = value;
	}

	public int getValue() {
		return this.value;
	}
}
public class TestEnum {
	public static void main(String[] args
		for(Day day:Day.values()) {
			// Line 1
		}
	}
}
```



What should be placed at line 1 to get the output as shown below?

SUNDAY-MONDAY-TUESDAY-WEDNESDAY-THURSDAY-FRIDAY-SATURDAY-

- ○ Not possible, the days will be displayed with their values only
- ○ System.out.print(day.getValue()+"-");
- ◉ System.out.print(day.name ()+"-");
- ○ System.out.print(day.getName()+"-");

What will be the output of the code given below?

```
import java.util.*;

public class Test {
        public static void main(String[] args) {
                List<Long> l1 = new ArrayList<Long>();
                l1.add(Long.valueOf(10));
                List<Number> l2 = l1;
                System.out.println(l2);
        }
}
```

- [10]
- Runtime Exception
- Compile time error
- It would print the reference object address

```
import java.util.*;

public class ListImpl {
        public static void main(String[] args) {
                List<String> al = new ArrayList<String>();
                al.add("infosys");
                al.add("google");
                al.add("ibm");
                al.add("Amazon");
                for (int j = 0; j < al.size(); j++) {
                        al.remove(j);
                        if (al.get(j++).equals("google"))
                                al.add("Oracle");
                        }
                }
                System.out.println(al);
        }
}
```

What is the output?

- [google, Oracle, ibm]
- [google, ibm, Oracle]
- [google, Amazon, Oracle]
- Runtime Exception

What is the output of the following code?

```
class Employee {
        public static void display() { // line
                System.out.print(" Employee ");
        }
}

public class Trainee extends Employee {
        public static void display() { // lin
                System.out.print(" Trainee ");
        }
        public static void main(String[] args
                Employee employee = new Trainee
                employee.display(); // line 9
        }
}
```

- ○ Trainee
- ◉ Employee
- ○ compile error at line 5
- ○ compile error at line 9

What will happen if the following code is compiled and executed?

(Choose 2 options)

```
package com.mypack.demoOne;
public class Employee {
	int id = 1000;
	protected String name = "Smith";
	public String role = "Software Engineer";
}

package com.mypack.demoTwo;
import com.mypack.demoOne.*;
public class Trainee {
	public static void main(String[] args
		Employee emp = new Employee();
		System.out.print(" "+ emp.id);
		System.out.print(" " + emp.name
		System.out.print(" "+ emp.role)
	}
}
```

Warning

This operation is

- [x] 1000 Smith Software Engineer
- [x] Compilation error at line 11
- [x] Compilation error at line 12
- [x] Compilation error at line 13
- [ ] Compilation error at line 14

What is the output of the below code?

```
public class Tester1 {
	public static void main(String[] args) {
		String s1 = "Infosys";
		String s2 = "Infosys";
		if (s1 == s2) {
			System.out.print("Equal");
		} else {
			System.out.print(" UnEqu
		}
		if (s1.equals(s2)) {
			System.out.print(" Equal
		} else {
			System.out.print(" UnEqu
		}
	}
}
```

- ○ Equal Unequal
- ◉ Equal Equal
- ○ UnEqual Equal
- ○ UnEqual UnEqual

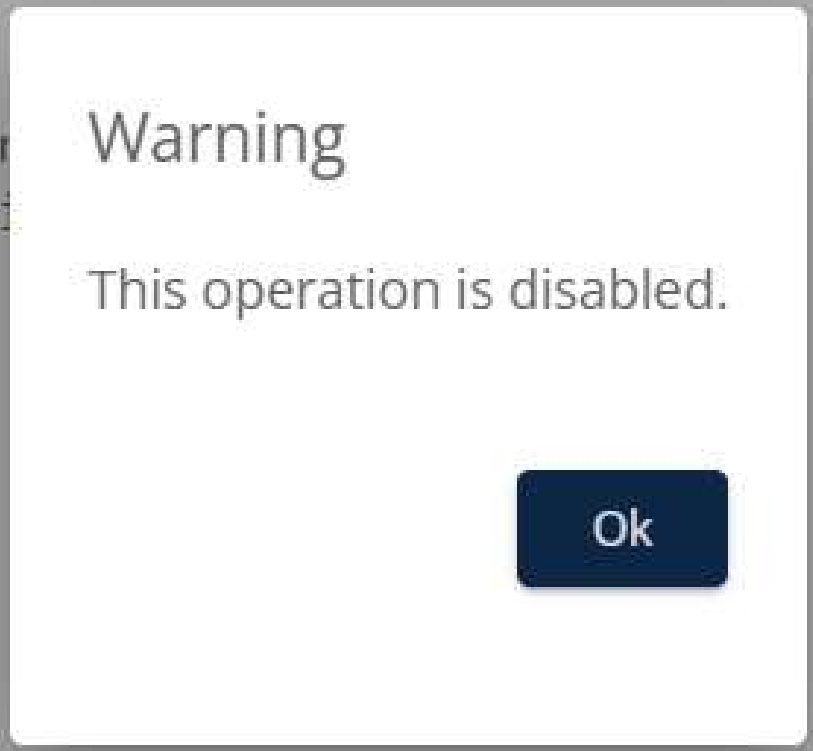
What is the output of the following code?

```
abstract class Employee {
   private void display() {
         System.out.print(" Employee ");
   }
}

public class Trainee extends Employee {
   protected void display() { // line 5
        System.out.print(" Trainee ");
   }
   public static void main(String[] args)
         Employee emp = new Trainee();
         emp.display(); // line 10
   }
}
```

- ○ Employee
- ○ Trainee
- ◉ Compiler error at line 5
- ○ Compiler error at line 10

Predict the output of the following code

```
public class VarargsTest {
            public static void main(String[] args) {
                    new VarargsTest().display(5,"Infosys");
                    new VarargsTest().display(4,"Infosys","Limited");
            }

          public void display(int b, Strin
                    System.out.print(stri
      }
}
```



- (●) Infosys Limited
- ( ) Infosys Infosys Limited
- ( ) Infosys
- ( ) Infosys Limited Infosys

Find the statements which are true with respect to method overriding

i. Method signature should vary (no of arguments and its type)

ii. Method to be invoked is decided at runtime

iii. Method to be invoked is decided based on the object

iv. Method can be static and private

- ○ i, ii
- ○ ii, iii and iv
- ◉ ii, iii
- ○ i, iii and iv

Warning

This operation is disab

Ok

What is the result of attempting to compile and run the program?

```
public class InfyTest {
	public static void main(String[] args) {
		int x, y, z;
		System.out.println(x + y + z);
	}
}
```

Warning

This operation i

- ○ Prints: null
- ○ Prints: 0
- ○ Run-time error, since x, y and, z is not initialized
- ◉ Compile-time error, since x, y and, z is not initialized

What is the result of attempting to compile and run the program?

```
public class StringTest {
	public static void main(String[] args) {
		String s1 = "A", s2 = "a", s3 = "b";
		s1.toLowerCase();
		s3.replace('b', 'a');
		System.out.print((s1.equals(s2)
	}
}
```

Warning

This operatic

- (●) Prints: false,false
- ( ) Prints: false,true
- ( ) Prints: true,false
- ( ) Prints: true,true

What is the output of the following code?

```
class Employee {
   public void display() {
          System.out.print(" display ");
   }

   public void print(int age) {
          System.out.print(" Employee ");
   }
}

public class Trainee extends Employee {
   public void display(String name) { // line 2
          System.out.print(name);
   }
   public int print(int age) { // line 5
          System.out.print(" Trainee ");
          return age;
   }
   public static void main(String[] args) {
          Trainee trainee = new Trainee();
          trainee.display(); // line 10
          trainee.display("Bob"); // line 11
          trainee.print(10); // line 12
   }
}
```

- (●) display Bob Trainee
- ( ) compile error at line 5
- ( ) compile error at line 10
- ( ) compile error at line 12

```
class Person{
        String name;
        Person(String name){
                this.name=name;
        }
        public boolean equals(Object ob){
                return ((Person)ob).name==this.name;
        }

        public int hashCode(){
                return name.length();
        }
}

public class MapImpl {
        public static void main(String[] args) {
                Map<Person,String> map=new HashMap<Person,String>()
                map.put(new Person("jack"),"sendSalesReport");
                map.put(new Person("jack"),"sendAuditReport");
                map.put(new Person("mady"),"sendInventoryReport");

                System.out.println(map.size());
        }
}
```

Wa

This

Which of the following is correct?(Choose 2 Options)

- [x] The output will be 3
- [ ] The output will be 2
- [x] If the hashcode method is commented the output will be 3
- [ ] If the hashcode method is commented the output will be 2
- [ ] If the hashcode method is commented the code will not compile

Select all possible options among the following:

Enums can be defined inside_____

(Choose 3 options)

- [x] An interface
- [x] A Class
- [x] A Static Context
- [ ] A Method

What is the output of the given code?

```
public class Tester {
	public static void main(String[] args) {
		Base obj = new Derived();
		obj.method();
	}
}

class Base {
	public void method(int a) {
		System.out.println("Base Metho
	}
}

class Derived extends Base {
	public void method() {
		System.out.println("Derived Me
	}
}
```

Wa

This

- (●) Compilation Error
- ( ) Base Method
- ( ) Derived Method
- ( ) Runtime Exception

Consider the following code:

```
void main(String args[])
{
        System.out.print("This is without class");
}
```

Predict the output of above code?

- ◯ Program will compile without any error and t class"
- ◯ NoSuchOperation Exception is thrown.
- ◉ Compilation Error as the name of the Java Cl
- ◯ Exception thrown "Cannot find main class".

Can you add custom methods to a record in Java 21?

- ○ Yes, but only instance methods
- ◉ Yes, both static and instance methods
- ○ No, records cannot have any methods
- ○ Yes, but only static methods

Warning

This operatic

Consider the following switch expression:

```
String result = switch (obj) {
    case String s -> s.toUpperCase();
    case Integer i -> i.toString();
    default -> "Unknown";
};
```

What will be the value of result if obj is a String obj

W

Th

- ○ "Unknown"
- ○ An exception will be thrown
- ○ "hello"
- ◉ "HELLO"

Which of the following is NOT a characteristic of a record in Java 21?

- ○ Automatically generated equals and hashCo
- ◉ Support for inheritance
- ○ Compact representation in memory
- ○ Implicit final fields

Warning

This operation

What is the purpose of the $ symbol in String Templates?

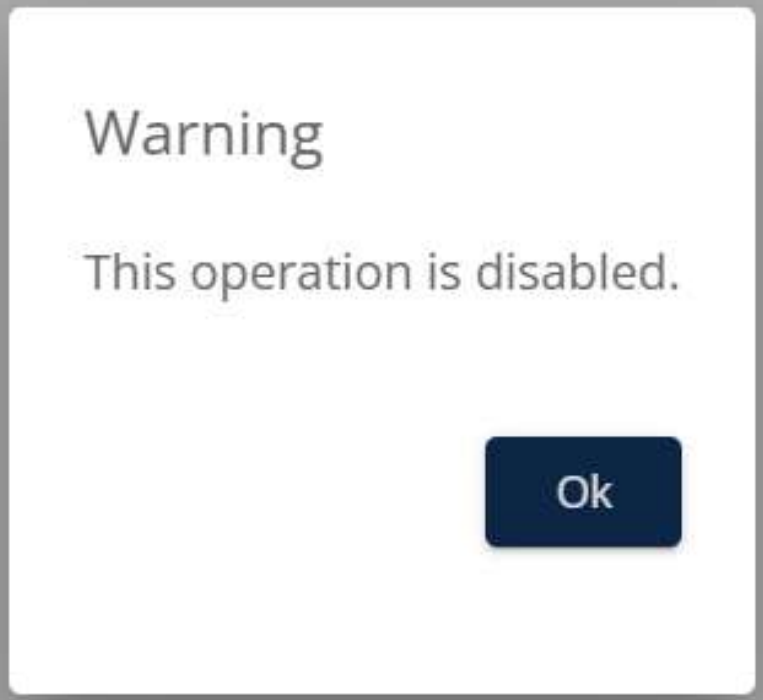
- (●) To indicate the start of a variable or express
- ( ) To denote the end of a String Template.
- ( ) To escape special characters within the temp
- ( ) To specify the data type of the inserted valu

Warn

This op

Consider the following code written inside main() method. Predict the output of the following code?

```
LinkedHashMap<String, String> m = new LinkedHashMap<>();
m.putFirst("First","1st");
m.putLast("Last", "lst");
System.out.println(m.sequencedEntrySet());
 System.out.println(m.sequencedKeySet());
System.out.println(m.reversed());
```



- [First,lst]
  [First, Last]
  {Last=lst, First=1st}
- {Last=lst, First=1st}
  [First=1st, Last=lst]
  [First, Last]
- [First, Last]
  [First=1st, Last=lst]
  {Last=lst, First=1st}
- [First=1st, Last=lst]
  [First, Last]
  {Last=lst, First=1st}

What is the default access modifier for an unnamed class?

- (●) Package-private
- ( ) protected
- ( ) private
- ( ) Public

Warnir

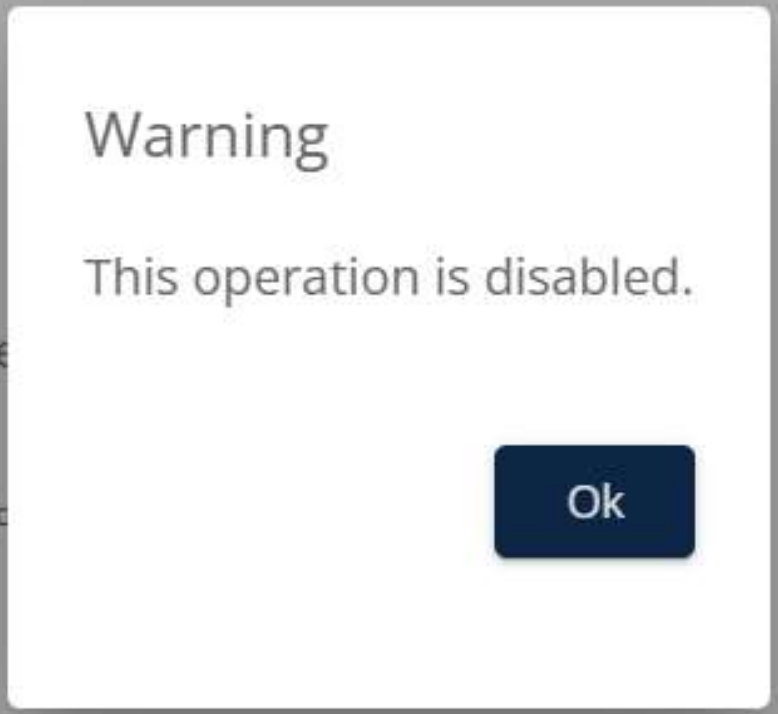
This ope

Consider the following method written inside a class.

```
static String getObjectsSharepriceWithSwitchAndUnnamedPattern(Object object)
 {
 return switch (object)
 {
 case Company(_, String shares, _) -> shares;
 default -> "No shares!";
 };}
```

Predict the output of the following code when calle

```
String obj2 = "Hello";
getObjectsSharepriceWithSwitchAndUnnamedPattern(c
```



- ◯ LinkageError will be thrown
- ◯ NotSupportedException will be thrown
- ◯ shares
- ◉ Hello

Which of the following is a potential benefit of using unnamed patterns?

- (●) Improved code readability
- ( ) Reduced memory usage
- ( ) Increased performance
- ( ) Increase Complexity

Warning

This operation is c

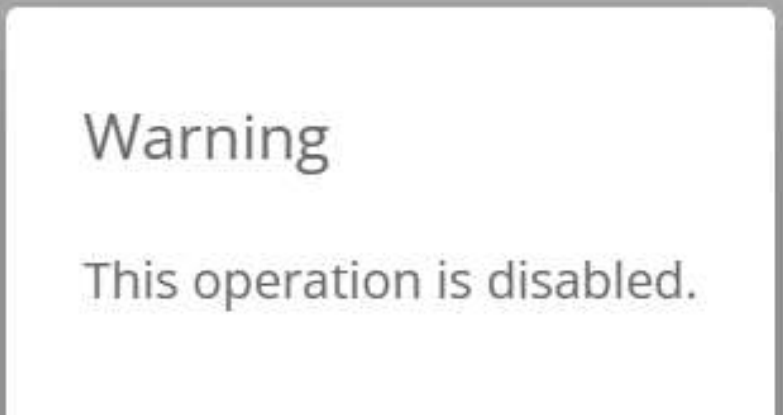
Predict the output of the following code snippet?

```
public class PatternMatching {

          static void test(Object o)
          {
          switch (o)
          {
          case null:
 System.out.println("OOps");
          break;
          case String s: System.out.println("Great");
          break;
          default: System.out.println("Ok");
           }
           }
          public static void main(String[] args) {

                String obj1 = "H";
                String obj2 = "Hello";
                String obj3 = null;

test(obj3);
        }
}
```

- (●) OOps
- ( ) Great
- ( ) NullPointerException
- ( ) Operation not allowed

Which of the following is a valid way to format a number with two decimal places in a String Template?

- ○ "${number:.2f}"
- ◉ "${number:2f}"
- ○ "${number:0.2f}"
- ○ "${number:02f}"



Which of the following interfaces represent Sequence Collections in Java 21?

- ○ List and Set
- ○ Map and Queue
- ◉ SequencedCollection, SequencedSet, and Se
- ○ Iterable and Collection

Warning

This operation is disabl