Moving to another question will save this response.

Question 1 of 35

**Question 1**

0.5 points Saved

Infinite recursion occurs when: (choose all that apply)

- [x] the base case is never reached by the recursive step.
- [x] the algorithm contains no base case.
- [x] when the recursive case is never reached by the base case.
- [ ] the method contains two or more base cases.

Moving to another question will save this response.

Question 1 of 35

Question Completion Status:

This is a matching question: for each of the following recursive methods, identify the problem with the recursion, if any, by matching it with a **single** issue from the provided list:

G.
```
// call from main: example1(5);
int example1(int n) {
 int a=10;
 if (n==0) return 0;
 return example1(n+1);
 }
```

G.
```
// call from main: example2(10);
int example2(int n) {
 int a=10;
 if (a == 100) return 1;
 return example2(n-1) + 1/n; }
```

D.
```
// call from main: example3(8);
int example3(int n) {
 int a;
 if (n<=0) return 1;
 else {
 a = (example3(n-2) + 3*example3(n/2) + 5);
 return a;
 }
```

F.
```
// call from main: example4(6);
int example4(int n) {
 int a=10;
 return example4(n-1) + 1;
 if (n == 0) return 1;
 }
```

A. Missing recursive case.

B. Too many base cases.

C. The base case is missing.

D. No issues, recursion is correct.

E. Too many recursive cases.

F. Improper placement of the base case.

G. The recursive case does not approach the base case.

Moving to another question will save this response.

Question 2 of 35

**Question 3**

0.25 points Saved

After inserting the following elements in a stack: 5, 1, 3, 9, 7. The element at the top of the stack will be [B1]

7

Moving to another question will save this response.

Question 3 of 35

Moving to another question will save this response.

Question 4 of 35

**Question 4**

0.25 points Saved

Given the following definitions:

```
class Vehicle <T>{}
class Car <T>extends Vehicle<T> {}
```

which of the following assignments is legal?

- `Vehicle <Object> b = new Vehicle <String> ();`
- `Vehicle <Object> b = new Car <String> ();`
- `Vehicle <String> b = new Car <String> ();` (selected)
- None of the above.

Moving to another question will save this response.

Question 4 of 35

Moving to another question will save this response.

Question 5 of 35

**Question 5**

0.75 points Saved

1. Consider the following recursive method:

```
public static boolean p(String s, int i, int f){
   if (i < f) {
      if (s.charAt(i) == s.charAt(f)) {
         return p(s, i+1, f-1);
      } else {
         return false;
      }
   } else {
      return true;
   }
}
```

What is the output for each of the following statements?

| | |
|---|---|
| 1. `System.out.println(p("ABDBA", 0, 4));` | true |
| 2. `System.out.println(p("ABCA",0, 3));` | false |
| 3. `System.out.println(p("AAA",0, 2));` | true |

**Question 8**

0.25 points Saved

Given the following method declaration:

```
public static int f( int x, int y ) {
    if( x == 0 )
        return y;
    else return f( x - 1, y + 1 );
    }
```

what value is returned as the result of the `call f (3,9)`?

- 1
- 0
- The recursion is incorrect
- 12 (selected)

Moving to another question will save this response.

Question 8 of 35

Moving to another question will save this response.

Question 12 of 35

**Question 12**

0.25 points Saved

Given the following class hierarchy :

```
public abstract class A {...}
public class AB extends A {...}
public class BC extends AB {...}
```

If we declare the following array in a test class:

```
A[] array = new A[100];
```

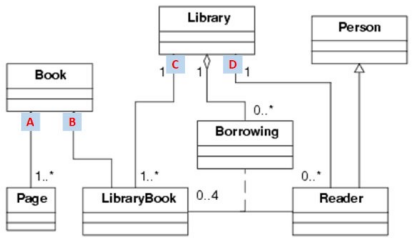
The array can store:

- (•) Instances of class AB and BC.
- ( ) T he array initialization is incorrect
- ( ) Instances of class A and AB.
- ( ) Instances of class A only.

Moving to another question will save this response.

Question Completion Status:

Given the following altered UML diagram:




Decide the type of relationship that should appear between the classes of the following UML diagram:, by choosing the correct relation type:

2. ▾ A
3. ▾ B
4. ▾ C
3. ▾ D

1. Association
2. Composition
3. Aggregation
4. Inheritance
5.

Moving to another question will save this response.

Question 17 of 35

**Question 17**

0.25 points Saved

The declaration of the following generic class is considered legal.

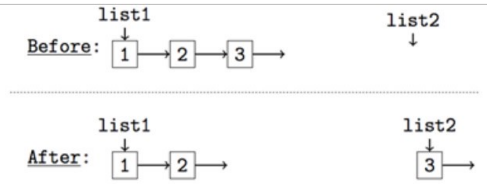
```
public class Box<Integer, String> {...}
```

- (•) True
- ( ) False

Moving to another question will save this response.

Question 17 of 35

Question Completion Status:




```
list1.next = null;
list2 = list1.next.next;
```

```
list2 = list1.next.next;
list1.next = null;
```

```
list2 = list1.next.next;
list1.next.next = null;
```

```
list2 = list1.next;
list1.next = null;
```

**Question 19**

1 points Saved

Match each of the following terms with its definition:

A. ▾ Abstraction

B. ▾ Polymorphism

C. ▾ Inheritance

D. ▾ Encapsulation

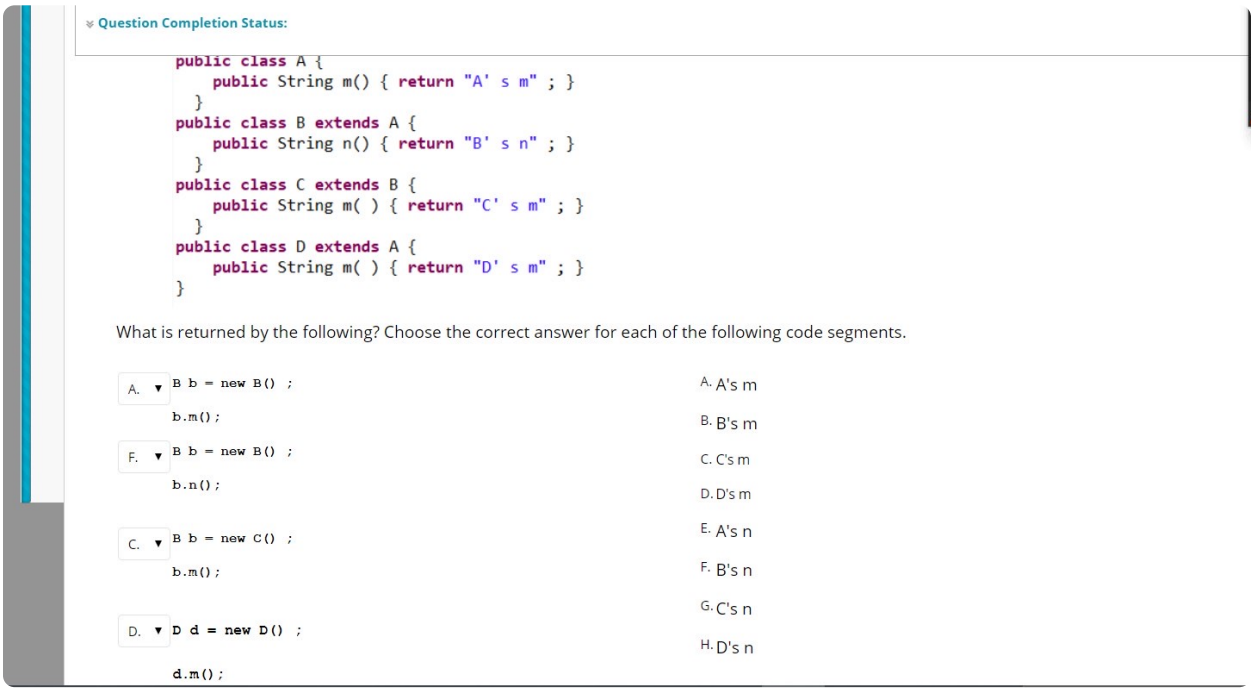
A. The use of only a public interface of a class in order to represent its methods to other classes; to represent a class only by its external behavior.

B. The ability of an object to take on the form of different classes depending on how it is used, especially in the instance of a subclass standing in for an instance of its parent class. It also refers to the ability to adapt the behaviors of a class as it descends from other classes

C. The ability of new classes to be created by taking all existing data and behaviors of an existing class and extending it with new methods and data.

D. The combining of data and the behaviors which act on it into a single unified data structure, such that they can behave as a self-sufficient object and so that implementation details can be hidden away from the user.

⚠ Moving to another question will save this response.




Choose ( more than one) methods should be used when implementing Queue using a linked list?

1) insertAtBack removeFromFront() ✅
2) insertAtFront() removeFromFront()
3) insertAtFront() removeFromBack()
4)

What's the output?

```
public static void mystery(int x, int y){

if(x>y)
System.out.print("*");
else if (x==y)
        System.out.print(y);
        else {
         System.out.print(y+ " ");
         mystery(x+1,y-1);
         System.out.print(" "+x);}}
    public static void main(String[] args) {
        mystery(1,4);}
}
```

Choose more than one:
super can be used to access instance members (variables/methods) in an immediate parent class
super can be used to invoke the constructor of an immediate parent class
super can be used to access instance members (variables/methods) in a grandparent class
super can be used to invoke the constructor of any grandparent class

Write a single line of code to create a node n with value 10 and the “next” node set to the node containing 3 in x such as shown in the following diagram:

```
int binarySearch(A, int K) { // A in an array of integer
  int left=0; right=n-1;  while left ≤ right do
    int middle=floor((left+right)/2)
if  (K == element at index 'middle' in A)
return middle
else if (K < element at index 'middle' in A)
right = middle-1
else left  = middle+1
return -1       }
```

```
class Node<T>{
T data;
Node<T> next;}
class NodeList<T> {
    Node<T> firstNode;
    public void method1(T value){
        Node<T> temp=new Node<T>();
        temp.data=value;
        temp.next=firstNode;
        firstNode=temp;}

    public T method2(){
        if (firstNode==null) return null;
        T temp =firstNode.data;
        firstNode=firstNode.next;
        return temp;}

    }
}
```

Choose : stack , queue , linkedlist
used to store and restore a recursive method and its arguments during the implementation of a recursive call. “Stack”
Typically used to implement a waiting mechanism for a set of processes sharing the same resource, such that the first process added would be the first to access the resource. “ queue”
The data structure in which we must keep track of two links. linked list

We like to define a hospital system hierarchy. This hierarchy is composed of following classes:
1) Patient : patient can be admitted in one room , patient can have many bills to pay.
2) Doctor : can treat from 5 to 15 patients
3) Department: have up to 20 doctors.
4) Room : can hold up to 3 patients .
Using this information , complete the following UML multiplicity by filling in the blanks with the appropriate values.

Edible is an interface and classes Salad and Fruit both implement Edible
Choose the legal ?
1) = new Edible () ;
2) = new Edible () ;
3) = new Edible () ;
4) =new Salad() ✅

Choose one “ dependent on another object for its existence? “Compstion”

Line of unnecessary casting? 18 21
(1-) رجعي مالقيتيها واذا ظهور للقيمة لأول الاندكس جيبي valueو head ميثود وقالوا تستقبل ال
ميثود نفس اللي تو لكن رجعي عدد مرات ظهور القيمة
مع افتراض ان اللست != null واول قيمة لها الصفر

ANSWER ٣اوتبوت
null or ERORR or اللي طلع

الجافا تسمح بوراثة اكثر من انترفيس
True or false: <T,S> is legal? True