**Computer Programming II (CS141)**

**Sample Practice Question Paper**

**Q. 1. Multiple Choice Questions**

Q. 1.1. Which of the following statements are incorrect?

1. public members of class can be accessed by any code in the program.
2. private members of class can only be accessed by other members of the class.
3. **private members of class can be inherited by a sub class, and become protected members in sub class.**
4. protected members of a class can be inherited by a sub class, and become private members of the sub class.

Q. 1.2. A recursion occurs when \_\_\_

1. a constructor calls a method
2. **A method calls itself**
3. a method calls another method
4. A constructor calls another constructor.

Q. 1.3. Which is true about a method-local inner class?

1. It must be marked final.
2. **It can be marked abstract.**
3. It can be marked public.
4. It can be marked static.

Q. 1.4. The methods of interface are \_\_\_\_\_\_ by default.

1. **abstract**
2. static
3. final
4. none of these

Q. 1.5. Which is true about an anonymous inner class?

1. It can extend exactly one class and implement exactly one interface.
2. It can extend exactly one class and can implement multiple interfaces.
3. **It can extend exactly one class or implement exactly one interface.**
4. It can implement multiple interfaces regardless of whether it also extends a class.

Q. 1.6. Which of these data types is used by operating system to manage the Recursion in Java?

1. Array
2. **Stack**
3. Queue
4. Tree

Q.1.7. Which of these will happen if recursive method does not have a base case?

1. **An infinite loop occurs**
2. System stops the program after some time.
3. After 1000000 calls it will be automatically stopped.
4. None of the mentioned

Q.1.8. Consider square numbers defined as follows (for positive integers): ?

square(1) = 1

square(N) = square(N-1) + 2N -1

According to this definition, what is square(3)?

1. square(3) = square(2) + square(1)
2. square(3) = square(2) - 2\*3 +1
3. **square(3) = square(2) + 2\*3 -1**
4. square(3) = square(3) + 2\*3 -1

Q.1.9. The correct method for declaring an interface is:

1. **public interface InterfaceName{ methodSignatures();}**
2. public static interface InterfaceName{ methodSignatures();}
3. public static final interface InterfaceName{ methodSignatures();}
4. None of the above

Q.1.10. While searching in a sorted array which algorithm performs better

1. Linear Search
2. **Binary Search**
3. Sequential Search
4. Both Binary and Sequential take same time

Q.1.11 What is the worst case performance of insertion sort

1. **O(n2)**
2. O(n)
3. O(n log n)
4. O(log n)

Q.1.12. Suppose an array contains 1000000 elements. Using the binary search algorithm, one require only about n comparisons to find the location of an item in the array, then n is:

1. 40 b) 30 **c) 20** d) 10

Q. 1.13. The running time of an algorithm is given by T(n) = 8T(n/2) +qn, n > 1, where q is constant and n is the input size. The Order of algorithm is:

1. n2 b) nn **c) n3** d) n

Q. 1.14. Sorting is useful for

1. Report generation
2. Responding to queries easily
3. Making searching easier and efficient
4. **All of these**

**Q. 1.15. You want to check whether a given set of items is sorted. Which of the following sorting methods will be most efficient if it is already in sorted order?**

1. Selection Sort
2. **Insertion Sort**
3. Merge Sort
4. Quick Sort

**Q. 1.16. Which of these packages contain all the collection classes?**

1. java.lang
2. **java.util**
3. java.net
4. java.awt

Q. 1.17. Which statement is true for the Class java.util.HashSet?

1. A: The collection is guaranteed to be immutable.
2. **B: The elements in the collection are unique.**
3. C: The elements in the collection are ordered.
4. D: The elements in the collection are synchronized.

**Q. 1.18. Which interface does java.util.Hashtable implement?**

1. **Java.util.Map**
2. Java.util.List
3. Java.util.HashTable
4. Java.util.Collection

**Q. 1.19. Which interface provides the capability to store objects using a key-value pair?**

1. Java.util.HashTable
2. Java.util.Set
3. Java.util.List
4. **Java.util.Map**

**Q. 1.20. Which statement is true for the class java.util.HashSet?**

1. The elements in the collection are ordered.
2. The collection is guaranteed to be immutable.
3. **The elements in the collection are guaranteed to be unique.**
4. The elements in the collection are accessed using a unique key.

**Q. 1.21. Which statement is true for the class java.util.ArrayList?**

1. **The elements are ordered.**
2. The collection is guaranteed to be immutable.
3. The elements in the collection are guaranteed to be unique.
4. The elements in the collection are accessed using a unique key.

**Q. 1.22. Which of the following is false statement?**

1. The Iterator interface declares only three methods: hasNext, next and remove.
2. **The ListIterator interface extends both the List and Iterator interfaces.**
3. The ListIterator interface provides forward and backward iteration capabilities.
4. The ListIterator interface provides the ability to modify the List during iteration.

**Q. 1.23. Which of the following statements about the hashcode() method are incorrect?**

1. The value returned by hashcode() is used in some collection classes to help locate objects.
2. **The hashcode() method is required to return a positive int value.**
3. Two new empty String objects will produce identical hashcodes.
4. None of the above

**Q. 1.24. Which of the following is Java reserved word?**

1. run
2. imports
3. **default**
4. implement

**Q. 1.25. Which of the following statements are true about Iterator?**

1. The Iterator interface is used to step through the elements of a Collection.
2. Iterators let you process each element of a Collection.
3. Iterators are a generic way to go through all the elements of a Collection no matter how it is organized
4. **All of the above.**

**Q. 2. Write True or False.**

|  |  |  |
| --- | --- | --- |
| 1 | Interface types do not make code reusable | FALSE |
| 2 | Java programming language supports eight primitive data types. | TRUE |
| 3 | Compiler turns an inner class into a regular class file | TRUE |
| 4 | The constructor overloading is similar to method overloading in Java | TRUE |
| 5 | Protected data cannot be accessed by all methods of classes in the same package  | FALSE |
| 6 | Inheritance is a mechanism for extending existing classes by adding instance variables and methods | TRUE |
| 7 | Objects of a super class can always be assigned to a subclass reference.  | FALSE |
| 8 | Polymorphism is the ability to treat objects with differences in behavior in a uniform way | TRUE |
| 9 | Association uses specific relationship between classes | FALSE |
| 10 | **A recursive computation solves a problem by using the solution of the same problem with simpler values** | TRUE |

**Q. 3. Write Short Answers for the following questions**

**Q. 3.1.** **Suppose you sign a contract, promising that you will, for an agreed-upon price, design, implement, and test a software package exactly as it has been specified in a requirements document. What is the primary risk you and your customer are facing with this business arrangement?**

**Answer:** It is unlikely that the customer did a perfect job with the requirements document. If you don’t accommodate changes, your customer may not like the outcome. If you charge for the changes, your customer may not like the cost.

**Q. 3.2. Why would an inner class method want to access a variable from a surrounding scope?**

**Answer:** Direct access is simpler than the alternative — passing the variable as a parameter to a constructor or method..

**Q.3.3. What is difference between List and a Set?**

1. List can contain duplicate values but Set doesnt allow. Set allows only to unique elements.
2. List allows retrieval of data to be in same order in the way it is inserted but Set doesnt ensures the sequence in which data can be retrieved.(Except HashSet)

**Q.3.4. Why wouldn’t you want to use a stack to manage print jobs?**

**Answer:** Stacks use a “last in, first out” discipline. If you are the first one to submit a print job and lots of people add print jobs before the printer has a chance to deal with your job, they get their printouts first, and you have to wait until all other jobs are completed.

**Q.3.5. You can compute the factorial function either with a loop, using the definition that n! = 1 × 2 × ... × n, or recursively, using the definition that 0! = 1 and n! = (n - 1)! × n. Is the recursive approach inefficient in this case?**

**Answer:** No, the recursive solution is about as efficient as the iterative approach. Both require n - 1 multiplications to compute n!.

**Q.4. What is the output of this program?**

import java.util.\*;

class stack {

 public static void main(String args[]) {

 Stack obj = new Stack();

 obj.push(new Integer(3));

 obj.push(new Integer(2));

 obj.pop();

 obj.push(new Integer(5));

 System.out.println(obj);

 }

}

**output: [3,5]**

**Q.5.1. Draw a Binary Search Tree by inserting the following keys from left to right:**

13, 3, 4, 12, 14, 10, 5, 1, 8, 2, 7, 9, 11, 6, 18

**Solution:**



**Q. 5.2. Draw a new heap after deleting the root from the following heap.**



**Solution: After deleting root:**



**Q.4. Write a program to sort the following array using selection sort algorithm**

**int[] arr = (10,34,2,56,7,67,88,42)**

Solution:

Using Selection Sort

public class SelectionSort {

 public static int[] doSelectionSort(int[] arr) {

 for (int i = 0; i < arr.length - 1; i++) {

 int index = i;

 for (int j = i + 1; j < arr.length; j++) {

 if (arr[j] < arr[index]) {

 index = j;

 }

 }

 int smallerNumber = arr[index];

 arr[index] = arr[i];

 arr[i] = smallerNumber;

 }

 return arr;

 }

 public static void main(String a[]) {

 int[] arr1 = {110, 34, 2, 56, 7, 67, 88, 42};

 int[] arr2 = doSelectionSort(arr1);

 for (int i : arr2) {

 System.out.print(i);

 System.out.print(",");

 }

 }

}