

Q1: For each of the following multiple choice questions, choose one correct answer.

1. An interface type is similar to a class, but there are several important differences:
 - a) All methods in an interface type are abstract; they don't have an implementation.
 - b) All methods in an interface type are automatically public.
 - c) An interface type does not have instance fields. *variable*
 - d) All of the above.
2. Which of the following is an event source?
 - a) A JButton object.
 - b) An event listener.
 - c) An inner class.
 - d) An event adapter.
3. Assume we are using quicksort to sort an array in ascending order. What can we conclude about the indexes of two consecutively placed pivot elements?
 - a) They are randomly located.
 - b) They are in different halves of the array.
 - c) They are both in the same half of the array.
 - d) They are adjacent.
4. ____ is often described as the *is-a* relationship.
 - a) Inheritance.
 - b) Aggregation.
 - c) Polymorphism.
 - d) Dependency.
5. After 1 iteration of selection sort working on an array of 10 elements, what must hold true?
 - a) the array cannot be sorted
 - b) the largest element is correctly placed
 - c) at least two elements are correctly placed
 - d) one element must be correctly placed

6. Identify the terminating condition in the following code.

```
int add(int i, int j) { // assumes i >= 0
    if (i == 0)
        return j;
    else
        return add(i - 1, j + 1);
}
```

- a) if (i == 0)
 - b) return j
 - c) return add(i - 1, j + 1)
 - d) there is no terminating condition
7. Assuming that parameter n is non-negative, what is the recursive condition (which allows the recursive method to be called) of method mystery?

```
public int mystery (int n, int m) {
    if (n == 0)
        return 0;
    if (n == 1)
        return m;
    return m + mystery(n - 1, m);
}
```

- a) $n == 0$
 - b) $n == 1$
 - c) $n > 0$
 - d) $n >= 2$
8. A linear search locates a value in an array of length n in ____ steps.
- a) $O(n^2)$
 - b) $O(\log n)$
 - c) $O(n)$
 - d) $O(\log n^2)$
9. A subclass can access all its superclass variables if they are declared:
- a) private or public.
 - b) protected or public.
 - c) public only.
 - d) protected only.



10. The performance of an algorithm is most closely related to what?

- a) the total number of element visits
- b) the total number of elements
- c) the type of elements
- d) the number of lines of code in the method

[10 Marks]

Q2: For each of the following statements, answer with True or False.

1.	Aggregate relationship is stronger form of dependency.	T
2.	Just like inheritance a class can implement <u>only one</u> interface.	F
3.	A recursive a method is simply a method that calls itself.	T
4.	Selection sort is better than merge sort for a large data set.	F
5.	The classical Waterfall method for software development is characterize by agility and <u>flexibility</u> ?	F

[5 Marks]

Q3: Give a brief answer for each question below.

1. If an element is present in an array of size n, how many element visits, in the worst case, are necessary to find it using a linear search? Why?

N

The maximum visits for a linear search of an array of n elements is n because assuming the element is located at last location.

And the linear search locates a value in an array in $O(n)$ steps

[2 Marks]

2. What do you do if a CRC card has ten responsibilities?

Reword the responsibilities so that they are at a higher level, or come up with more classes to handle the responsibilities

2

[2 Marks]



3. What steps does the selection sort algorithm go through to sort the sequence 3 1 4 6 2 5?

Note: there may be more rows than needed, filling any extra row may cause your mark to be reduced!

3	1	4	6	2	5
3 ✓	3 ✓	4 ✓	6 ✓	2 ✓	5 ✓
1 ✓	2 ✓	4 ✓	6 ✓	3 ✓	5 ✓
1 ✓	2 ✓	3 ✓	6 ✓	4 ✓	5 ✓
1 ✓	2 ✓	3 ✓	4 ✓	6 ✓	5 ✓
1 ✓	2 ✓	3 ✓	4 ✓	5 ✓	6 ✓

[4 Marks]

4

4. What does the User Interface Event do? And define the term Event Listeners.

- User interface *events* include key presses, mouse moves, button clicks, and so on • Most programs don't want to be flooded by boring events . A program can indicate that it only cares about certain specific events
- **Event listener:**
 - *Notified when event happens*
 - *Belongs to a class that is provided by the application programmer*
 - *Its methods describe the actions to be taken when an event occurs*

5. List (in order) the different phases of the development process for the software life cycle.

1-Analysis:

- Decide what the project is supposed to do. Do not think about how the program will accomplish tasks.

2-Design

Plan how to implement the system. Discover structures that underlie problem to be solved. Decide what classes and methods you need.

3-Implementation

• Write and compile the code. • Code implements classes and methods discovered in the design phase . Completed program.

4-Testing: Run tests to verify the program works correctly. Program Run: A report of the tests and their results.

5-Deployent: Users install program. Users use program for its intended purpose.

Q4: Suppose an algorithm takes 12 seconds to handle a data set (n) of size 800. Fill in the following table, which shows the approximate growth (in seconds) of the execution times depending on the complexity of the algorithm.

Note 1: You MUST show your calculations.

Note 2: Round your results to the nearest decimal point.

	$O(n^2)$	$O(n \log n)$
5000	469.2 469.2	96
10000	1875.6 1875.6	206.4 206.4

$$O(n^2) \Rightarrow \frac{(5000)^2}{(800)^2} = 39.1$$

$$39.1 \times 12 \text{ s} = \cancel{468} \text{ s} \rightarrow 469.2 \text{ s}$$

$$O(n^2) \Rightarrow \frac{(10000)^2}{(800)^2} = 156.3$$

$$156.3 \times 12 = \cancel{1875.6} \text{ s} \rightarrow 1875.6 \text{ s}$$

$$O(n \log n) \Rightarrow \frac{5000 \log(5000)}{800 \log(800)} = 8$$

$$8 \times 12 \text{ s} = 96 \text{ s}$$

$$O(n \log n) \Rightarrow \frac{10000 \log(10000)}{800 \log(800)} = 17.2$$

$$17.2 \times 12 \text{ s} = \cancel{206.4} \text{ s} \rightarrow 206.4 \text{ s}$$

8

Q5: Read the following program carefully and produce its exact expected output.

```
class A
{
    A()
    {
        System.out.println("This is class A");
    }
    public void a()
    {
        System.out.println("This is method a");
    }
}

class B extends A
{
    B()
    {
        Take a constructor from A class super();
        System.out.println("This is class B");
    }
    public void b()
    {
        System.out.println("This is method b");
    }
}

class C extends B
{
    C()
    {
        Take a constructor from class B and B super();
        System.out.println("This is class C");
    }
    public void c()
    {
        System.out.println("This is method c");
    }
}
```

main
cb = new C();

Continues next page ...

```
class Main
{
    public static void main (String args[])
    {
        C b = new C();
    }
}
```

Output:

This is class A ✓
This is class B ✓
This is class C ✓
~~This is method C~~

12