Final Quiz

Due 10 Jun at 16:40 **Points** 40 **Questions** 30
Available 10 Jun at 15:10 - 10 Jun at 16:45 1 hour and 35 minutes
Time limit 90 Minutes

Instructions

Final Quiz - Week 13

There are **30 questions** in this quiz. It will last for 90 mins.

Please, note it is a closed book quiz, so no electronics, browsing or compiler allowed.

This quiz is no longer available as the course has been concluded.

Attempt history

	Attempt	Time	Score
LATEST	Attempt 1	89 minutes	34 out of 40

Score for this quiz: **34** out of 40
Submitted 10 Jun at 16:39
This attempt took 89 minutes.

Question 1 **1 / 1 pts**

Consider the following code fragment with initial values a = 30, b = -10, ans = 5

```
if (a >= 10) {
    ans += 10;
    if (a < 30) {
        ans = 30;
    } else if (b > 0){
        ans = a - ans + 30;
    } else {
        ans++;
    }
} else {
    ans = a;
    if (b <= 20) {
        ans = a - ans + b;
    }
}
```

What's the value of *ans* after running this code?

- 50
- 30
- Correct! 16
- 11

Question 2

1 / 1 pts

The following Java declarations have been made:

```
int a = 10;
int b = a - 20;
boolean c = false;
```

What is the value of the following boolean expression

```
(b <= a) && (a + b < 0) || !c && (b - a > b)
```

True

Correct!

False

Question 3

1 / 1 pts

Select the best description for what this mystery function calculates

```
public int mystery(int numbers[]) {
 int maximum = 1;
 int count = 1;
 for ( int i=1 ; i<numbers.length ; ++i ) {
   if ( numbers[i-1] == numbers[i] )
     count++;
   else
      count = 1;
   if ( maximum < count )
      maximum = count;
  }
return (maximum);
}
```

the number of times any two consecutive elements of the array are equal

the length of the longest consecutive set of numbers in the array that are the maximum value

the number of times an element matches the first value

the number of times the maximum value occurs in the array

Correct!

the length of the longest consecutive set of numbers in the array that are equal

Question 4

1 / 1 pts

You have a 2D array with 6 rows and 3 columns. What is the correct way to access the last (rightmost) element of the last row?

- `array[6][0]`
- `array[2][5]`
- Correct! `array[5][2]`
- `array[6][3]`

Question 5

0 / 1 pts

Read the following method skeleton and choose the best expression to fill in the blank on **line 8** so that the method will behave correctly:

```
/**
* Takes a string reference and counts the number of times
* the character 'A' or 'a' appears in the string object.
* @param aString   String reference to object containing chars.
* @precondition    aString is not null (you may assume this is tr
ue).
* @return          The number of times 'A' or 'a' appears in the
string.
*/
public static int countAs(String aString)  // line 1
{
    int counter = __________;              // line 2
    int totalA = 0;                        // line 3
    while (counter < __________)           // line 4
    {
        if ( __________.equals("A") )      // line 5
        {
            totalA = totalA + __________;  // line 6
        }
        counter++;                         // line 7
    }
    return __________;                     // line 8
}
```

- true

u Answered

- counter

rrect answer

- totalA
- false

Question 6

0 / 1 pts

Suppose you want to write a program that reads in numbers provided by a user, until the user types a negative value. Then, the program outputs the largest value it has read from the user.

Your program will

ıu Answered

- needs two loops and one array
- needs an array and one variable.

ırrect answer

- need a loop but does not need an array.
- needs two arrays and a while loop
- need one loop and one array.

Question 7

1 / 1 pts

In our program, we define the Animal class as

```
class Animal{
    // other code ...

    public Animal() {
        this.name = "unknown";
        this.age = -10;
        this.species = "unknown";
    }

    public Animal(String name, String species) {
        this.name = name;
        this.species = species;
    }

    public Animal (String name, int age) {
        this.name = name;
        this.age = age;
    }

    public printInfo() {
        System.out.println("Name: " + this.name + "; species: " +
this.species
                                   + "; age: " + this.age);
    }

    // other code ...
}
```

What is the output if we run the following scripts

```
Animal crow = new Animal("Roy", 4);
Animal tiger = new Animal("Bruce", "tiger");
crow.printInfo();
tiger.printInfo();
```

```
Name: Roy; species: crow; age: 4
Name: Bruce; species: tiger; age: 0
```

```
Name: Roy; species: null; age: 4
Name: Bruce; species: tiger; age: null
```

Correct!

```
Name: Roy; species: null; age: 4
Name: Bruce; species: tiger; age: 0
```

```
Name: Roy; species: unknown; age: 4
Name: Bruce; species: tiger; age: -10
```

Question 8

2 / 2 pts

What's the value of *array* after running the following code?

```
int[][] array = new int[3][4];

// Initialize all elements of array with zeros

for (int i=0; i<2; i++) {
    for (int j=0; j<4; j++) {
        array[i][j] = i + j*2;
        if (j == 3) {
            break;
        }
    }
}
```

Correct!

```
{{0, 2, 4, 6},
 {1, 3, 5, 7},
 {0, 0, 0, 0}}
```

```
{{0, 2, 4, 0},
 {1, 3, 5, 0},
 {0, 0, 0, 0}}
```

```
{{0, 1, 2, 3},
 {2, 3, 4, 5},
 {4, 5, 6, 7}}
```

```
{{0, 1, 2, 3},
 {2, 3, 4, 5},
 {0, 0, 0, 0}}
```

Question 9

1 / 1 pts

What is an **incorrect** way to overload methods?

```
String myfunc(double par1, int par2);
int myfunc(String newPar1, String newPar2);
```

```
int myfunc(double par1, int par2);
int myfunc(double par1, String par2, int par3);
```

```
String myfunc(double par1, int par2);
int myfunc(double par1);
```

Correct!

```
String myfunc(double par1, int par2);
int myfunc(double newPar1, int newPar2);
```

Question 10

2 / 2 pts

Which access modifier means that the member can be accessed only within this class, its subclasses and other classes in the same package?

Correct!

- protected
- private
- public

Question 11

1 / 1 pts

Assume we have defined the following classes. What is wrong here?

```
abstract class AbstractVehicle {
    public String type;

    public AbstractVehicle() {
        this.type = "unknown";
    }

    public void setType(String type) {
        this.type = type;
    }

    public abstract void printInfo();
}
```

```
interface Drivable {
    public String color;
    public void drive();
}
```

```
public class Car extends AbstractVehicle implements Drivable {
    private int speed;

    public void printInfo() {
        System.out.println("This is a car");
    }

    public void drive() {
        this.speed = 100;
    }
}
```

- Drivable interface must extend AbstractVehicle class
- AbstractVehicle cannot have a constructor
- AbstractVehicle cannot have non-abstract method setType
- Car class cannot implement an interface, because it already inherits from abstract class

Correct!

Drivable interface cannot have non-static attribute color

Question 12

2 / 2 pts

The Linux command interpreter or shell is the program users interact with in a terminal emulation window. In this task, you are asked to match a list of shell command with their descriptions.

Correct!	**pwd**	To know which path/di
Correct!	**ls**	To know what files are
Correct!	**mv file1.txt ..**	To move file to parent (
Correct!	**mv file1.txt file2.txt**	To rename file
Correct!	**cd ~**	To go to home director

Other Incorrect Match Options:

- To delete files and directories
- To go to parent directory
- To move file to home directory

Question 13

1 / 1 pts

Assume we have defined the following classes:

```
class Person {
    public void print() {
        System.out.println("This is Person");
    }
}
```

```
class Student extends Person {
    public void print() {
        System.out.println("This is Student");
    }
}
```

```
class Doctor extends Person {
    public void print() {
        super.print();
        System.out.println("This is Doctor");
    }
}
```

What's the output when we run the following code?

```
// other codes...
Person student = new Student();
Person doctor = new Doctor();
student.print();
doctor.print();
// other codes...
```

- ```
  This is Person
  This is Student
  This is Doctor
  ```
- ```
  This is Person
  This is Person
  ```
- ```
  This is Person
  This is Student
  This is Person
  This is Doctor
  ```
  ```

Correct!

```
This is Student
This is Person
This is Doctor
```

Question 14

1 / 1 pts

Consider that you have three algorithms with different Big O notation. The first runs in $O(n^2)$ time complexity, the second runs in $O(n)$ and the third runs in $O(n*log(n))$.

In this task, you are asked to analyze which algorithm need **LESS** steps when $n>100000$ (asymptotically large).

- Depends on the computer running on
- For large n values the time is almost the same
- Correct! O(n)
- O(n^2)
- O(n*log(n))

Question 15

1 / 1 pts

The *calculate* method is defined below

```
public static int calculate(int num) {
    if(num % 5 == 0){
        return 1;
    } else if (num == 8){
        return -3;
    }
    return 1 + calculate(num+2);
}
```

What's the return value when you call

```
calculate(2);
```

- -3
- 2
- 4

Correct!

- 0

Question 16

2 / 2 pts

This program calculates recursively the number of trucks needed to carry a load on numcrates.

How many recursive calls are made <u>in total </u>(excluding the initial call given here)   for numTrucks(9, 3) ?

```
public int numTrucks(int numCrates, int loadSize) {
        if (numCrates <= loadSize)
                return 1;
        int left = (numCrates + 1) / 2;
        int right = numCrates / 2;
        return numTrucks(left, loadSize) + numTrucks(right, loadS
ize);
    }
```

- 3

Correct!

- 6
- 8
- 5

Question 17

1 / 1 pts

What sentence is true about the pivot in Quicksort?

- We must use the last element for the pivot

Correct!

- After partitioning, the pivot will never move again
- Before partitioning, it is always the smallest element in the list

After partitioning, the pivot will always be in the centre of the list

A random choice of pivot is always the optimal choice, regardless of input

Question 18 **0 / 2 pts**

We have a computing system where comparison is done very fast and doesn't affect the running time much. But swapping (or assignment) is extremely time-consuming operation. Which sorting algorithm would be a better choice given the random input?

ɩu Answered

Insertion

Bubble

ɩrrect answer

Selection

Question 19 **1 / 1 pts**

What is the best case time complexity of Bubble Sort?

Correct!

O(n)

O(log(n))

- $O(n^2)$
- O(n*log(n))

Question 20

2 / 2 pts

Suppose you try to perform a binary search on the unsorted array {1, 4, 3, 7, 15, 9, 24}.

Which element will not be found when you try searching for it?

- 24
- none of them
- 7
- 1
- Correct! 15

Question 21

2 / 2 pts

What are the advantages of LinkedList comparing to an array? You can select more than one

- Correct! Inserting elements is faster in LinkedList
- Array needs extra memory to store the indices

Accessing elements is faster in LinkedList

Correct!

Deleting elements is faster in LinkedList

The size of LinkedList is fixed

Question 22

1 / 1 pts

Suppose q is an instance of a queue that can store Strings, and I execute the following statements starting with q empty:

```
q.enqueue("Sweden");
q.enqueue("is");
q.enqueue("my");
String w = q.dequeue();
String x = q.peek();
q.enqueue("neighbour");
String y = q.dequeue();
String z = q.dequeue();
```

What is the final value of z ?

"neighbour"

none of the above

"is"

Correct!

"my"

Question 23

1 / 1 pts

Which of the following abstract datatypes would be the best choice for part of the implementation of the part of a compiler that determines whether the parentheses in a program are balanced?

- a tree

Correct!

- a stack
- a queue
- a linkedList

Question 24

1 / 1 pts

Consider the following implementation of the 'deque' method in Queue class. Assume that classes Queue and Node are implemented correctly.

```
1: public void dequeue() {
2:     if (this.rear == null) {
3:         return;
4:     } else {
5:         Node tmp = this.rear;
6:         while (tmp.next.next != null) {
7:             tmp = tmp.next;
8:         }
9:         tmp.next = null;
10:    }
11:}
```

Which line can cause an error? You only need to enter the number.

Correct!

6

rrect Answers

6 (with margin: 0)

Question 25

0 / 2 pts

Locating a new node's insertion point in a binary search tree stops when

- We find a node lesser than the node to be inserted.
- We reach the tree's maximum level.
- rrect answer — We reach a null child.
- u Answered — We find a node without any children.

Question 26

1 / 1 pts

What is the **worst** time complexity to search a value in Binary Search Tree?

- $O(n^2)$
- O(log(n))
- Correct! — O(n)
- O(n*log(n))

Question 27

2 / 2 pts

Given the following binary tree, how many comparisons we need to perform to insert number 14 into this tree?

```
          10
        /     \
      4         15
    /          /  \
  1          11   21
```

Correct!

- (•) 3
- ( ) 2
- ( ) 4
- ( ) 5

Question 28

1 / 1 pts

Which of the following is **incorrect**?

Correct!

- (•) Each node in a graph can have only one parent
- ( ) Each node in a tree can have multiple children
- ( ) Tree is a graph with specific properties
- ( ) Graph can have loops

Question 29

1 / 1 pts

Given a graph saved as an array of adjacency lists:

```
Node 0: 0 -> 2
Node 1: 2 -> 3
Node 2: 0 -> 3
Node 3: 1
```

What will be the corresponding adjacency matrix?

	Node 0	Node 1	Node 2	Node 3
Node 0	1	0	1	1
Node 1	0	0	1	1
Node 2	1	1	0	1
Node 3	1	1	1	0

	Node 0	Node 1	Node 2	Node 3
Node 0	1	0	1	0
Node 1	0	0	1	1
Node 2	1	1	0	1
Node 3	0	1	1	0

	Node 0	Node 1	Node 2	Node 3
Node 0	1	0	1	1
Node 1	0	0	1	0
Node 2	1	0	0	1
Node 3	0	1	0	0

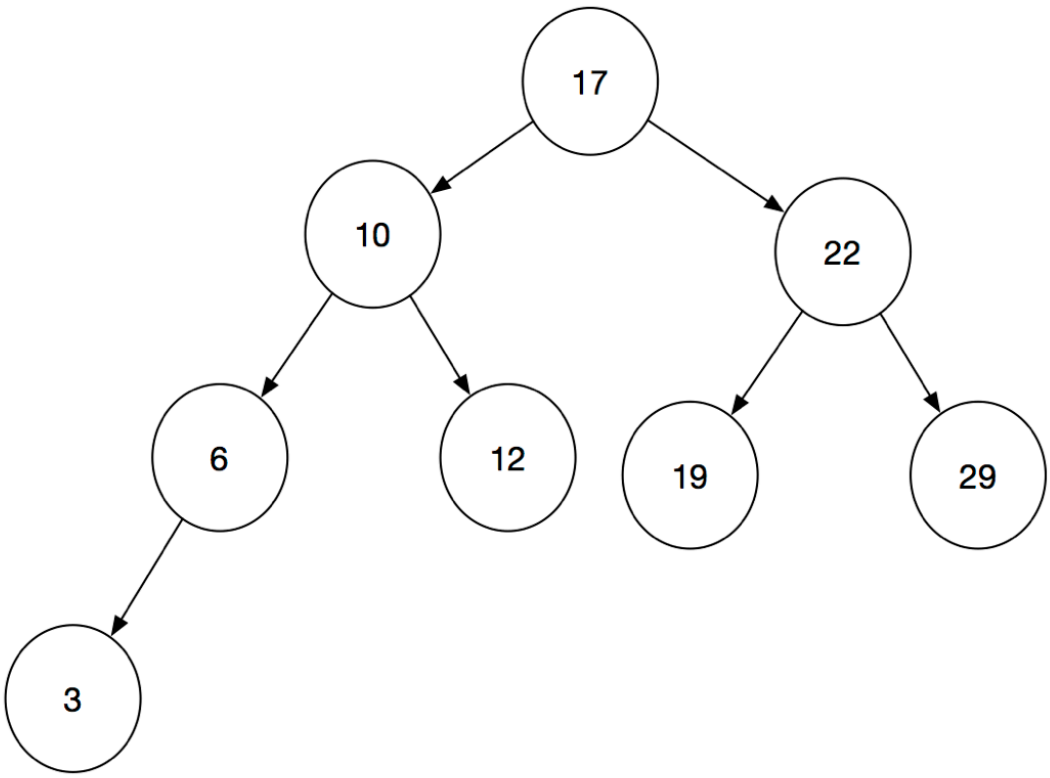
Correct!

	Node 0	Node 1	Node 2	Node 3
Node 0	1	0	1	0
Node 1	0	0	1	1
Node 2	1	0	0	1
Node 3	0	1	0	0

Question 30

2 / 2 pts

The order of printing the nodes by **Pre-Order** traversal of the following tree is



3, 6, 12, 10, 19, 29, 22, 17

Correct!

17, 10, 6, 3, 12, 22, 19, 29

3, 6, 10, 12, 17, 19, 22, 29

17, 10, 6, 12, 3, 22, 19, 29

Quiz score: **34** out of 40