



University of Cyprus
Department of Electrical and
Computer Engineering

ΜΑΘΗΜΑ / COURSE : HMMY 318 / ECE 318
ΗΜΕΡΟΜΗΝΙΑ / DATE: 17-12-2023
ΩΡΑ / TIME: 12:30

ΟΔΗΓΙΕΣ: Παρακαλώ απαντήστε **ΟΛΕΣ** τις ερωτήσεις. Η εξέταση είναι **ΑΤΟΜΙΚΗ** και όχι ομαδική.

Απαγορεύεται η χρήση οποιασδήποτε μορφής βοήθημα από το Διαδίκτυο, υπολογιστική μηχανή, ηλεκτρονικό υπολογιστή, έξυπνα κινητά ή έξυπνα ρολόγια. Παρακαλώ όπως φυλάξετε όλες τις έξυπνες συσκευές σας.

INSTRUCTIONS: Please answer **ALL** questions. This is an **INDIVIDUAL** piece of assessment and NOT group-based.

You are not allowed to use any help from the Internet through any smart devices such as smart phones, smart watches, laptops or calculators. Please safely store any smart device prior starting this exam.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ / NAME & SURNAME:

Αριθμός ταυτότητας/ student ID:

Question 1: General Programming Properties / Memory Management (20 Marks)
Ερώτηση 1: Γενικές Ιδιότητες Προγραμματισμού / Διαχείριση Μνήμης (20 Μονάδες)

1. Please name the main components of the Von Neumann computer architecture.
Παρακαλώ ονομάστε τα βασικά στοιχεία στην αρχιτεκτονική υπολογιστών του Von Neumann.

(2 marks/μονάδες)

2. State the sequence of the three steps associated to the execution of a Java program as written by a programmer according to the Java programming language environment.
Ονομάστε τα τρία βασικά βήματα εκτέλεσης για ένα πρόγραμμα γραμμένο στην Java όπως είναι καθορισμένα από το περιβάλλον της Java.

(2 marks/μονάδες)

3. State the two key differences between the C programming language and the Java programming language.
Αναφέρετε τις δύο βασικές διαφορές μεταξύ της γλώσσας προγραμματισμού C και της Java.

(2 marks/μονάδες)

4. Explain what the Java Virtual Machine (JVM) is and why it is important.
Εξηγήστε τι είναι το Java Virtual Machine (JVM) και γιατί είναι σημαντικό.

(4 marks/μονάδες)

5. Explain how primitive and non-primitive data types are addressed and stored in the Java programming language.
Εξηγήστε πως οι primitive και non-primitive τύποι δεδομένων απευθύνονται και αποθηκεύονται στην μνήμη.

(4 marks/μονάδες)

6. Within 4 regular sentences:

- a) State the names of the three algorithmic approaches for garbage collection in Java.
- b) Explain the most recent algorithmic approach for garbage collection in Java.

Μέσα σε 4 κανονικές προτάσεις:

- a) Ονομάστε τους τρεις βασικούς αλγόριθμους για garbage collection στην Java.
- b) Εξηγήστε περιληπτικά τον πιο πρόσφατο αλγόριθμο για garbage collection στην Java.

(6 marks/μονάδες)

Question 2: Data Structures & Algorithms in Java (30 Marks)

Ερώτηση 2: Δομές Δεδομένων & Αλγόριθμοι στην Java (30 Μονάδες)

1. Provide the code required in the class below to provide the output: *"The result of comparing str1 and str2 is: true"*.

Συμπληρώστε τον κώδικα που χρειάζεται στην πιο κάτω κλάση ούτως ώστε να έχω την εκτύπωση: *"The result of comparing str1 and str2 is: true"*

(4 marks / μονάδες)

Code snippet:

```
class Main {  
    public static void main(String[] args) {  
        String str1 = "I love Jimi Hendrix";  
        String str2 = " I love Jimi Hendrix";  
  
        //code required to be included below:  
    }  
}
```

```
}  
}
```

2. Fill in the code snippet in the below class requiring a **while**, **for** and a **do-while** loop having a print output of:

```
Pearl Jam Live in Madison Square Garden, NY!  
Pearl Jam Live in Madison Square Garden, NY!  
Pearl Jam Live in Madison Square Garden, NY!  
Bob Dylan Live in Armadillo, Glasgow!  
Bob Dylan Live in Armadillo, Glasgow!  
Bob Dylan Live in Armadillo, Glasgow!  
Radiohead Live in O2 Arena, London!  
Radiohead Live in O2 Arena, London!
```

Συμπληρώστε τον κώδικα που χρειάζεται στην πιο κάτω κλάση χρησιμοποιώντας **while**, **for** και **do-while** loops ούτως ώστε να έχω την εκτύπωση:

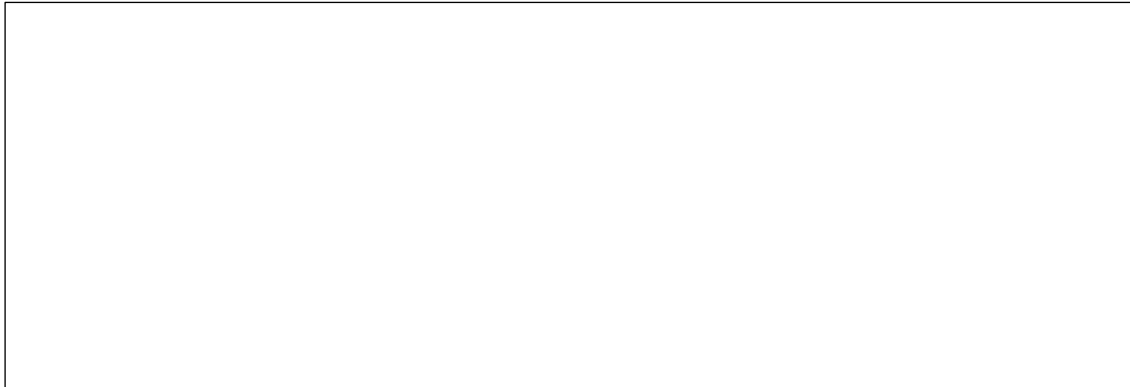
```
Pearl Jam Live in Madison Square Garden, NY!  
Pearl Jam Live in Madison Square Garden, NY!  
Pearl Jam Live in Madison Square Garden, NY!  
Bob Dylan Live in Armadillo, Glasgow!
```

```
Bob Dylan Live in Armadillo, Glasgow!  
Bob Dylan Live in Armadillo, Glasgow!  
Radiohead Live in O2 Arena, London!  
Radiohead Live in O2 Arena, London!
```

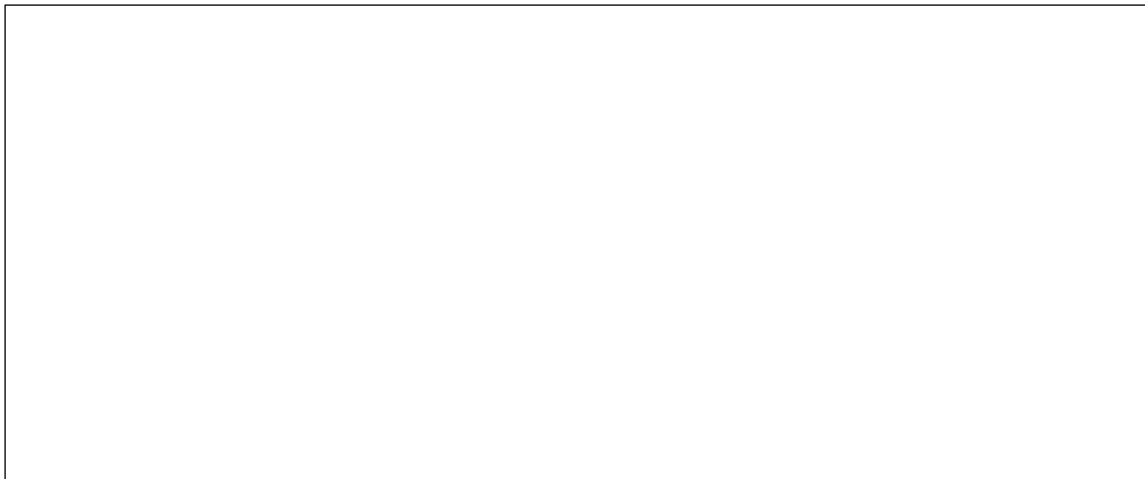
(10 marks/μονάδες)

Code snippet:

```
public class Adverts{  
    public static void main(String[] args) {  
  
        //complete while loop for Pearl Jam below
```



```
        // complete a for loop for Bob Dylan below
```



```
        // complete a do-while for Radiohead below
```

```
}  
}
```

3. Fill in the code snippet in the below class using a `for-each` loop to produce the printout:

```
20  
90  
50  
100
```

Συμπληρώστε τον κώδικα στην πιο κάτω κλάση χρησιμοποιώντας `for-each` loop για να έχει ως αποτέλεσμα την εκτύπωση:

```
20  
90  
50  
100
```

(4 marks / μονάδες)

Code snippet:

```
class Main {  
    public static void main(String[] args) {  
  
        // create an array  
        int[] nums = {2, 9, 5, 10};  
  
        // complete the for each loop below
```

```
}  
}
```

4. Fill in the code snippet in the below class using a loop to provide the following output:

```
0  
-1  
9  
-10  
-5  
2  
9  
10  
-2  
15
```

Συμπληρώστε τον κώδικα που χρειάζεται στην πιο κάτω κλάση χρησιμοποιώντας ένα βρόχο (loop) ούτως ώστε να εκτυπωθεί το πιο κάτω:

```
0  
-1  
9  
-10  
-5  
2  
9  
10  
-2  
15
```

(6 marks/μονάδες)

Code snippet:

```
public class IterateArray {  
    public static void main(String[] args) {
```



```
int[][] my2D = {  
    {0, -1, 9},  
    {-10, -5, 2, 9},  
    {10, -2, 15},  
};
```

//complete the required loop below

```
    }  
}
```

5. Based on the following code snippet of a *for* loop:
- State how many times the loop is executed.
 - Write the output of the program.
 - Convert the code into a nested *while* loop.

Με βάση το ακόλουθο απόσπασμα κώδικα ενός βρόχου τύπου *for*:

- Δηλώστε πόσες φορές εκτελείται ο βρόχος.
- Γράψτε τι θα εκτυπώσει το πρόγραμμα.
- Μετατρέψτε τον κώδικα σε ένθετο βρόχο *while*.

(6 marks / μονάδες)

Code snippet

```
for (int i=3; i<=4; i++){  
    for (int j=2; j<=i; j++){  
        System.out.println("");  
    }  
    System.out.println("Depeche Mode");  
}
```

a.

b.

c.

Question 3: Algorithmic Complexity, Searching and Sorting (20 Marks)

Ερώτηση 3: Αλγοριθμική Πολυπλοκότητα, Αναζήτηση και Ταξινόμηση (20 Μονάδες)

1. Which are the two main types of complexity we assess in algorithmic analysis and how we profile the various system, network, and architectural properties?

Ποιες είναι οι δύο βασικές μορφές πολυπλοκότητας που επικεντρώνεται η ανάλυση αλγορίθμων και πως συμπεριλαμβάνουμε τις συμπεριφορές του συστήματος, δικτύου και αρχιτεκτονικής του συστήματος;

(2 marks / μονάδες)

2. Considering the following code snippet:

- a) State its type of complexity in Big(O) notation.
- b) Convert it to $O(\log n)$.

Χρησιμοποιώντας τον πιο κάτω κώδικα:

- a) Ονομάστε σε ποια τάξη πολυπλοκότητας ανήκει σύμφωνα με το Big(O).
- b) Μετατρέψτε τον κώδικα ούτως ώστε να έχει απόδοση $O(\log n)$.

(3 marks / μονάδες)

Code snippet:

```
int i, n = 10;
for (i = 1; i <= n; i++)
{
    System.out.println("Liverpool FC!\n");
}
```

a.

b. New code snippet to be completed below:

3. Complete the following method to achieve binary search in a recursive mode.
Συμπληρώστε τον κώδικα που χρειάζεται στην πιο κάτω μέθοδο για να επιστρέφεται αποτέλεσμα βάση του αλγορίθμου δυαδικής αναζήτησης με αναδρομική συμπεριφορά.
(5 marks / μονάδες)

Code snippet:

```
int binarySearch(int arr[], int left, int right, int x) {  
  
    //new code to be inserted below
```

```
// We reach here when element is not in the array arr[] based
on search key x
    return -1;
}
```

4. Considering large data arrays that need to be sorted, discuss which algorithm between QuickSort and Mergesort is best to use. Your answer should be justified in terms of both time and space complexity.

Λαμβάνοντας υπόψη μεγάλους πίνακες δεδομένων που πρέπει να ταξινομηθούν, συζητήστε ποιος αλγόριθμος μεταξύ QuickSort και Mergesort είναι καλύτερο να χρησιμοποιηθεί. Η απάντησή σας θα πρέπει να αιτιολογείται τόσο από την άποψη της αλγοριθμικής πολυπλοκότητας του χρόνου όσο και του χώρου.

(4 marks / μονάδες)

5. Suppose we are sorting an array of eight integers using QuickSort, and we have just finished the **first** partitioning with the array looking like this:

2, 5, 1, 7, 9, 12, 11, 10

State which value could have been the pivot and justify your answer based on the properties of the algorithm.

Ας υποθέσουμε ότι ταξινομούμε έναν πίνακα οκτώ ακεραίων χρησιμοποιώντας QuickSort και μόλις ολοκληρώσαμε την **πρώτη** κατάτμηση με τον πίνακα να μοιάζει με:

2, 5, 1, 7, 9, 12, 11, 10

Δηλώστε ποια τιμή θα μπορούσε να ήταν ο pivot και αιτιολογήστε την απάντησή της βάσει της συμπεριφοράς του αλγόριθμου .

(4 marks / μονάδες)

6. Consider an array `myArray = {5,4,3,2,1}`. Write all the steps (i.e., order of numbers in each step) to be produced if we want to sort this array based on the insertion sort algorithm. If each step costs 10 nanoseconds (*ns*) also provide the total cost in (*ns*).

Θεωρήστε έναν πίνακα `myArray = {5,4,3,2,1}`. Γράψτε όλα τα βήματα (δηλαδή την σειρά των αριθμών σε κάθε βήμα) που θα παραχθούν εάν θέλουμε να ταξινομήσουμε αυτόν τον πίνακα με βάση τον αλγόριθμο ταξινόμησης `insertion sort`. Υποθέτοντας ότι κάθε βήμα κοστίζει 10 νανοδευτερόλεπτα (*ns*) βρείτε το ολικό κόστος.

(2 marks / μονάδες)

Question 4: Object-Oriented Programming Principles & Web Applications (30 Marks)

Ερώτηση 4: Αρχές Αντικειμενοστραφή Προγραμματισμού & Διαδικτυακές Εφαρμογές (30 Μονάδες)

1. Explain what inheritance is, describe what is a subclass and what is a superclass, and give an example of use using incomplete Java code.

Εξηγήστε τι είναι η κληρονομικότητα (inheritance), τι είναι μια υποκλάση (subclass) και τι είναι μια υπερκλάση (superclass) και δώστε ένα παράδειγμα χρήσης με ημιτελή κώδικα Java.

(6 marks / μονάδες)

2. Explain the difference between the public, private and the protected Java language keywords.

Εξηγήστε την διαφορά μεταξύ των λέξεων-κλειδιών, public, private και protected στην γλώσσα προγραμματισμού Java.

(3 marks / μονάδες)

3. Complete the following code snippet where a definition of a Polygon interface with a method related to the area of a polygon and an incomplete main class that should be instantiating an object of type Rectangle. The output of the complete program should be: *"The area of this rectangle is 25"*.

Συμπληρώστε το ακόλουθο απόσπασμα κώδικα όπου υπάρχει ήδη ένας ορισμός μιας διεπαφής Polygon με ημιπλήρη μέθοδο για το εμβαδόν ενός πολυγώνου και μιας ατελούς κύριας κλάσης που θα πρέπει να δημιουργεί ένα αντικείμενο τύπου ορθογώνιο. Η έξοδος του πλήρους προγράμματος θα πρέπει να είναι: *"The area of this rectangle is 25"*.

(6 marks / μονάδες)

Code snippet

```
interface Polygon {  
    void getArea(int length, int width);  
}
```

```
//class Rectangle below
```

```
class Main {  
    public static void main(String[] args) {  
        //need to call a Rectangle r1 instance and get area below
```

```
    }  
}
```

4. Considering the code snippet below explain and justify the type of inheritance we have and explain and justify what the output would be.

Λαμβάνοντας υπόψη το παρακάτω απόσπασμα κώδικα, εξηγήστε και αιτιολογήστε τον τύπο της κληρονομιάς που έχουμε και επίσης εξηγήστε/αιτιολογήστε τι θα εκτυπωθεί.

(5 marks / μονάδες)

Code snippet

```
class C
{
    public void disp()
    {System.out.println("C");}
}

class A extends C
{
    public void disp()
    {System.out.println("A");}
}

class B extends C
{
    public void disp()
    {System.out.println("B");}
}

class D extends A
{
    public void disp()
    {System.out.println("D");}
    public static void main(String args[]){
        D obj = new D();
        obj.disp();
    }
}
```

5. Employ **method overloading** in the Mult class such as to create a method `multiply()` to achieve:
- Adding 3 integers
 - Adding two integers
 - Adding 2 double type of variables

As soon as Mult is complete, complete the main method of the class Main also provided below to produce the following printout:

```
Multiplication with 2 integers:8  
Multiplication with 3 integers:16  
Multiplication with 2 doubles:16.5
```

Χρησιμοποιήστε υπερφόρτωση μεθόδου στην κλάση Mult, όπως για να δημιουργήσετε μια μέθοδο `multiply()` για να επιτύχετε:

- Προσθήκη 3 ακεραίων
- Προσθήκη δύο ακεραίων
- Προσθήκη 2 double τύπων μεταβλητών

Μόλις ολοκληρωθεί το Mult, ολοκληρώστε την κύρια μέθοδο της κλάσης Main που παρέχεται επίσης παρακάτω για να δημιουργήσετε την ακόλουθη εκτύπωση:

```
Multiplication with 2 integers:8  
Multiplication with 3 integers:16  
Multiplication with 2 doubles:16.5
```

(6 marks / μονάδες)

Code snippet

```
class Mult {
```

```
    //multiplication method(s) to complete below
```

```
}
```

```
// Main class
```

```
class Main {
```

```
    public static void main(String[] args)
```

```
    {
```

```
        // Code to complete below for output
```

```
    }
```

```
}
```

6. Explain the following:

- a. What is a URI and how it is decomposed?
- b. What is REST, which are its 4 main operations?
- c. On which protocol and which operations of that protocol are the 4 REST operations based on?
- d. Which are the layers of the OSI model?

Εξηγήστε τα ακόλουθα:

- a) Τι είναι το URI και πώς αποσυντίθεται;
- b) Τι είναι το REST και ποιες είναι οι 4 κύριες λειτουργίες του;

- c) Σε ποιο πρωτόκολλο και σε ποιες λειτουργίες αυτού του πρωτοκόλλου βασίζονται οι 4 βασικές λειτουργίες REST;
- d) Ποια είναι τα επίπεδα του μοντέλου αναφοράς OSI;

(4 marks / μονάδες)

/*/END OF EXAM SCRIPT – ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ