



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)

Main components are: Control Unit, Arithmetic Logic Unit (ALU), Memory composed of data and instructions and Input/Output (I/O).

2. State the sequence of the 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)

1. Compilation using a Java compiler
2. Conversion to Java bytecode
3. Interpretation from the Java Interpreter

3. State in no more than two short sentences the two key differences between the C programming language and the Java programming language.
Αναφέρετε μέσα σε δύο μικρές προτάσεις τις δύο βασικές διαφορές μεταξύ της γλώσσας προγραμματισμού C και της Java.

(2 marks/μονάδες)

C is a procedural and compiled programming language whereas Java is Object-Oriented that is both interpreted and compiled.

4. Explain in two regular sentences what the Java Virtual Machine (JVM) is and why it is important.

Εξηγήστε μέσα σε δύο σύντομες προτάσεις τι είναι το Java Virtual Machine (JVM) και γιατί είναι σημαντικό.

(4 marks/μονάδες)

The JVM is like a virtual PC understanding the specific Instruction Set of the Java bytecode. The JVM is important since it enables cross-platform operability and independence; hence Java programs can run on any platform/computer.

5. Explain in two regular sentences how primitive and non-primitive data types are addressed and stored in the Java programming language.

Εξηγήστε μέσα σε δύο σύντομες προτάσεις πως οι primitive και non-primitive τύποι δεδομένων απευθύνονται και αποθηκεύονται στην μνήμη.

(4 marks/μονάδες)

- Primitive types are stored using *direct addressing* and referred to as stored “*by value*”
- Non-primitive types are variables of a reference type, stored via *indirect addressing* and referred to as stored “*by reference*”

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)

The three basic GC algorithms are:

- *Reference counting*
 - *Tracing* (e.g., *Mark-Sweep*)
 - *Generational garbage collection....*
-
- Generational Garbage Collection is An "***aging***" approach
 - Objects usually have a short life span
 - Heap is decomposed into "generations"
 - Smaller chunks of memory
 - Stratified in terms of their "age"
 - objects that are "active" and for how long
 - 3 heap memory categories:
 - Young generation
 - Old generation
 - Permanent generation

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 str1 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"*

(5 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:
```

```
        // comparing str1 with str2  
        boolean result = str1.equals(str2);  
        System.out.println("The result of comparing str1 and str2 is: " + result);
```

```
    }  
}
```

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
```

```
int i = 0;  
while (i < 3) {  
    System.out.println("Pearl Jam Live in Madison Square Garden, NY!");  
    i++;  
}
```

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

```
for (i=0; i<=2; i++)  
{  
    System.out.println("Bob Dylan Live in Armadillo, Glasgow!");  
}
```

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

```
int j=1;
do
{
    System.out.println("Radiohead Live in O2 Arena, London!");
    j++;
}while(j<3);
```

```
    }
}
```

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
```

(5 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
```

```
// for each loop
for (int number: nums) {
    System.out.println(number*10);
}
```

```
}
}
```

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
```

(10 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

```
for (int i = 0; i < my2D.length; ++i) {  
    for(int j = 0; j < my2D[i].length; ++j) {  
        System.out.println(my2D[i][j]);  
    }  
}  
  
}  
}
```

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;

(3 marks / μονάδες)

We assess algorithms using time and space complexity and the various assumptions are formalised under a constant variable.

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 $O(\log n)$.

(5 marks / μονάδες)

Code snippet:

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

a. $O(n)$

b. New code snippet to be completed below:

```
int i, n = 10;
for (i = 1; i <= n; i=i*2) {
    System.out.printf("Liverpool FC!\n");
}
```

3. Complete the following method to achieve complete return of recursive binary search.

Συμπληρώστε τον κώδικα που χρειάζεται στην πιο κάτω μέθοδο για να επιστρέφεται αποτέλεσμα βάση του αλγορίθμου αναδρομικής δυαδικής αναζήτησης.

(8 marks / μονάδες)

Code snippet:

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

```
        if (right >= left) {  
            int mid = left + (right - left) / 2;  
  
            // If the element is present at the middle itself  
            if (arr[mid] == x)  
                return mid;  
  
            // If element is smaller than mid, then it can only be present in left subarray  
            if (arr[mid] > x)  
                return binarySearch(arr, left, mid - 1, x);  
  
            // Else the element can only be present in right subarray  
            return binarySearch(arr, mid + 1, right, x);  
        }  
    }
```

```
    // 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 in a maximum of 4 short sentences 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 με το πολύ 4 σύντομες προτάσεις 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)

Answer is quicksort.

Although both Quicksort and Mergesort have an average time complexity of $O(n \log n)$, Quicksort is the preferred algorithm, as it has an $O(\log(n))$ space complexity. Mergesort, on the other hand, requires $O(n)$ extra storage, which makes it quite expensive for arrays.

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

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

1. Explain in less than 4 regular sentences what is inheritance, describe what is a subclass and what is a superclass, and give an example.

Εξηγήστε σε λιγότερο από 4 κανονικές προτάσεις τι είναι η κληρονομικότητα (inheritance), τι είναι μια υποκλάση (subclass) και τι είναι μια υπερκλάση (superclass) και δώστε ένα παράδειγμα χρήσης.

(6 marks / μονάδες)

Inheritance in Java is a concept that acquires the properties from one class to other classes; like a father to a son. A subclass is the one that inherits from a superclass whereas the superclass the one that provides its functionalities and methods to its child class – i.e., the subclass. An example is having a general class of bike providing the main functionalities of a Bike and then two children classes (i.e., subclasses) namely; RacingBike and ShoppingBike inherit these main Bike properties but they also have their own distinct characteristics. The RacingBike has an additional gear whereas ShoppingBike has a ringing bell. (here the example is open – could be textually could be using pseudocode)

2. Explain in less than 2 regular sentences the difference between the public, private and the protected Java language keywords.

Εξηγήστε σε λιγότερο από 2 σύντομες προτάσεις την διαφορά μεταξύ των λέξεων-κλειδιών, public, private και protected στην γλώσσα προγραμματισμού Java.

(3 marks / μονάδες)

- Public variables can be accessed by other classes.
- Private variables can only be access by their class.
- Protected variables are accessible in the same package and subclasses.

3. Complete the following code snippet where a definition of a Polygon interface with a method related to the area of a polygon and a semi-complete main class instantiating an object of type Rectangle is already present. 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 ημιπλήρους κύριας κλάσης που 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);  
}
```

//need to create class Rectangle below

```
class Rectangle implements Polygon {  
  
    // implementation of abstract method  
    public void getArea(int length, int width) {  
        System.out.println("The area of the rectangle is " + (length * width));  
    }  
}
```

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

```
        r1.getArea(5, 5);
```

```
    }  
}
```

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

Λαμβάνοντας υπόψη το παρακάτω απόσπασμα κώδικα, υποδεικνύετε τον τύπο της κληρονομιάς που έχουμε και επίσης υποδεικνύετε 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();
    }
}

```

Class A and B extends class C → Hierarchical inheritance
 Class D extends class A → Single inheritance
 Overall we have hybrid inheritance!

The output is : D

5. Employ **method overloading** in the Mult class such as to create a method `multiply()` to achieve:

- a. Adding 3 integers
- b. Adding two integers
- c. 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:

- a. Προσθήκη 3 ακεραίων
- b. Προσθήκη δύο ακεραίων
- c. Προσθήκη 2 διπλών 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 {

    // 2 integer multiplication method to complete below

```

```
// Method with 2 integer parameters
static int multiply(int a, int b)
{
    // Returns product of integer numbers
    return a * b;
}
```

// 2 double multiplication method to complete below

```
// With same name but with 2 double parameters
static double multiply(double a, double b)
{
    // Returns product of double numbers
    return a * b;
}
```

// 3 integer numbers multiplication method to complete below

```
// // Multiplication of 3 int numbers
static int multiply(int a, int b, int c)
{
    // Return product
    return a * b * c;
}
```

```
}
```

// Main class

class Main {

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

// Code to complete below for printouts

```

// Calling method by passing
// input as in arguments
    System.out.println("Mutiplication with 2 integers:" +
Mult.multiply(2, 4));
    System.out.println("Mutiplication with 3 integers:" +
Mult.multiply(2, 4, 2));
    System.out.println("Mutiplication with 2 doubles:" +
Mult.multiply(5.5, 3.0));

}
}

```

6. In brief sentences, explain or name the following:

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

Σε σύντομες προτάσεις, εξηγήστε ή ονομάστε τα ακόλουθα:

- Τι είναι το URI και πώς αποσυντίθεται;
- Τι είναι το REST και ποιες είναι οι 4 κύριες λειτουργίες του;
- Σε ποιο πρωτόκολλο και σε ποιες λειτουργίες αυτού του πρωτοκόλλου βασίζονται οι 4 βασικές λειτουργίες REST;
- Ποια είναι τα στρώματα του μοντέλου αναφοράς OSI;

(4 marks / μονάδες)

- URI stands for Unified Resource Identifier determining the unique identification for a resource on a web server and it is decomposed into UR-Name and UR-Locator
- REST is an architectural style for implementing web applications and its main fundamental operations are CRUD – Create, Read, Update, Delete
- CRUD in REST directly relates with the operations in HTTP, namely Post, Get, Put/Post, Delete
- In OSI: Physical Layer, Data link, Network, Transport, Session, Presentation, Application

/*/END OF EXAM SCRIPT – ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ