# OOP-SW-ESPE-2024-14539-Exam1

**Puntos totales** 37/51   ❓

Object Oriented Programming - Universidad de las Fuerzas Armadas ESPE
Computer Science Department
Software Engineering
NRC: 14539

------------------->> Enjoy you must have
            My young Padawan! <<-------------------

READ THE INSTRUCTIONS FIRST
Answer every theoretical question using this google form. In the exercises, please upload to the OOP course GitHub repository (Exams/lastname/unit1/question##/) your Netbeans Project and screen shots (png) of the program running. Also upload a zip ZIP ZiP ziP zip file with everything to this Google Form Evaluation where it is required. Also the answer to question 27 must be uploaded to the repository, the vpp file and a pdf/jpg file.

*1 de 1 puntos*

You must unlearn what you have learned
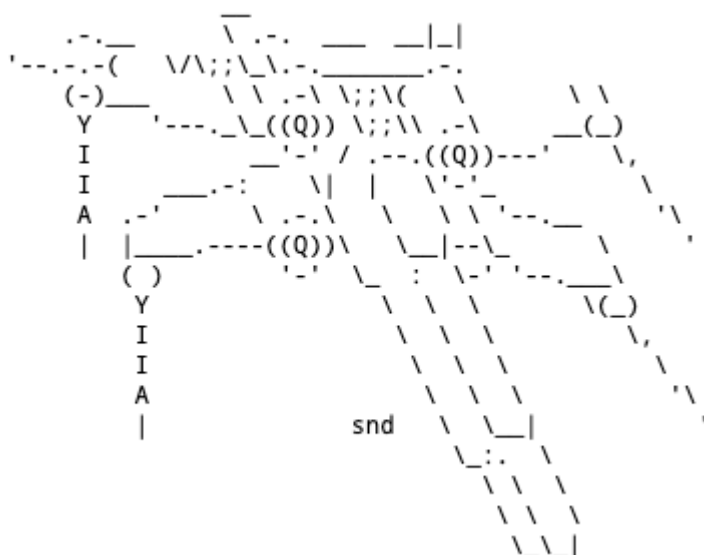


Enter your Full Name: LastName FirstName MiddleName, for example: Lascano Jorge Edison   *

...............................................................................................................................................

Enter your number on the roster (your list number) *

5

✓ Did you read the directions. If not, please do it. Por favor, lea las    *1/1
indicaciones antes de empezar su examen. Please make sure that your
computer is plugged in and your Internet is OK Por favor asegúrese de
tener electricidad conectada a su computadora. Y el Internet en buen
estado.

⦿ Yes                                                                        ✓

◯ No

**OOP Fundamentals (10 minutes)**                              **10 de 10 puntos**

✓ 1. It is used to model the structure of objects in the system *          1/1

⦿ Class Diagrams                                                             ✓

◯ Use Case Diagrams

◯ Flowcharts

✓ 2. Hiding of design decisions inside software appropriate components *    1/1

⦿ Encapsulation                                                             ✓

◯ Abstraction

◯ Inheritance

✓ **3. An object includes another object as subpart of it** *    **1/1**

○ Aggregation     ✓

○ Dependency

○ sub-object

✓ **4. Leave unnecessary definitions out of the system implementation** *    **1/1**

● Abstraction     ✓

○ Encapsulation

○ Objects

✓ **5. A class is composed of** *    **1/1**

○ Attributes and variables

● attributes and methods     ✓

○ functions and methods

✓ **6. The parent of all the classes in Java is known as** *    **1/1**

● Object     ✓

○ Class

○ object

○ class

✓  7. The creator of the Java Programming Language is *                    1/1

○  Oracle

○  Duke

●  James Gosling                                                            ✓

---

✓  8. Encapsulation is implemented by the keyword *                        1/1

○  public

○  void

●  private                                                                 ✓

---

✓  9. Aggregation, composition and association in a class diagram are      *1/1
   relationships between classes that are implemented in code using

●  attributes                                                             ✓

○  methods

○  packages

---

✓  10. Dependency in a class diagram, is a relationship between classes that *1/1
   are implemented in

○  variables

○  classes

●  methods                                                                ✓

GitHub skills (5 minutes)                                                           **3 de 5 puntos**

11. Match each of the following Git commands to its purpose *

| | clone | add | commit | push | pull | Puntuación | |
|---|---|---|---|---|---|---|---|
| Put one or more new, changed, or deleted files under version control | ○ | ● | ○ | ○ | ○ | 1/1 | ✓ |
| Create a replica of remote repository into folder on the local system | ● | ○ | ○ | ○ | ○ | 1/1 | ✓ |
| Make a new version | ○ | ○ | ● | ○ | ○ | 1/1 | ✓ |
| Sync any new versions on the local system with the remote system | ○ | ○ | ○ | ● | ○ | 0/1 | ✗ |
| Sync any new versions on the remote system with the local system | ○ | ○ | ○ | ○ | ● | 0/1 | ✗ |

**Respuestas correctas**

| | clone | add | commit | push | pull |
|---|---|---|---|---|---|
| Sync any new versions on | ○ | ○ | ○ | ○ | ● |

the local
system with
the remote
system

Sync any new
versions on
the remote            ○          ○          ○          ◉          ○
system with
the local
system

More OOP Fundamentals (5minutes)                              **5 de 5 puntos**

✓   12. What does it mean to instantiate a class object? *                1/1

○   duplicate a class

○   delete a class

◉   create an object from the class                                        ✓

○   connect two clases to each other

✓   13. A constructor is a *                                              1/1

○   variable

○   class

○   attribute

◉   method                                                                ✓

✓ 14. These diagrams help organize and model the requirements of a
system showing the cases of use and actors?

*1/1

○ sequence diagrams

○ collaboration diagrams

◉ use case diagrams ✓

○ class diagram

✓ 15. A getter is a *

1/1

◉ method ✓

○ attribute

○ class

○ variable

✓ 16. What name must have a constructor in Java? *

1/1

○ a verb

○ the name of the package

○ any name

◉ the same name as the class ✓

True or False (10 minutes)

8 de 10 puntos

✓ 17. Classification (the noun) is the process of group objects together into *1/1
sets based on common properties

○ True

◉ False ✓

---

✓ 18. Classification (the verb) or "class" is a set of objects that have the *1/1
same kinds of attributes and methods

○ True

◉ False ✓

---

✓ 19. One way to find potential classes in a system is to document a high- *1/1
level description of the system and look for nouns.  Those nouns are
most likely to represent meaningful classes.

◉ True ✓

○ False

---

✓ 20. Use cases are used to document the requirements (the goals) of a *1/1
system

◉ True ✓

○ False

✓  21. C++, C# and Java are structured programming languages *        1/1

○  True                                                                          ✓

◉  False                                                                         ✓

---

✗  22. Object Oriented code makes it easy to add new classes without        *0/1
   modifying existing functions

○  True

◉  False                                                                         ✗

Respuesta correcta

◉  True

---

✗  23. C ++ and Java are declarative programming languages *        0/1

◉  True                                                                          ✗

○  False

Respuesta correcta

◉  False

---

✓  24. A method that is called from another method inside the same class        *1/1
   should be defined later in the same class, i.e., after the method that called
   it

○  True

◉  False                                                                         ✓

✓   **25. WheelsList is a good name for a variable** *                              1/1

○ True

◉ False                                                                                    ✓

---

✓   **26. A good programming practice is to use nouns to name the methods** *   1/1

○ True

◉ False                                                                                    ✓

○ Otro: _____

Reverse Engineering skills, from code to UML (15 minutes)              **0 de 10 puntos**

✗  27. Classes + Relations (Reverse Engineering). Draw the class diagram  *.../10
   corresponding to the following code. Convert every attribute to
   associations, aggregations, compositions or dependencies, with
   appropriate names and multiplicity constraints (Upload the Png/Jpg file
   here, and the vpp and png file to the repository):

```
/** @author OOP|instructors */
public class POOExam1P27 {
  public static void main(String[] args) {
    E e = new E();Question10
    C c = new C();
    e.m2(c);
    //Optional code
    A a = new A();
    B b1 = new B();
    B b2 = new B();
    a.m1(b1, b2);
  }
}
public class A {
  private B b1;
  private B b2;
  /**
   * This method uses two objects of type B
   * @param x of type B
   * @param y of type B
   */
  public void m1(B x, B y){
  }
}
public class B {
}
public class C {
  private B b;
}
public class E {
  private A a;
  private B[] b = new B[10];
  /**
   * This method will allow to use an object of type C, and returns nothing
   * @param z this is an object of type C
   */

  public void m2(C z) {
  }
}
```

🖼 Class Diagram J...

⤒ Añadir archivo

**Comentarios**

*classes (attributes and methods) + relationships (dependencies, associations, aggregations, compositions, multiplicity constraints) -> 10 pts.*

Reading UML (5 minutes)                                    **10 de 10 puntos**

Based on the previous answer (Class diagram). Answer the following questions with an integer number. use digits. DON'T use words

✓   28. How many associations is the B class connected to? *                    2/2

3                                                                                  ✓

✓   29. How many associations is the A class connected to? *                    2/2

2                                                                                  ✓

✓   30. The multiplicity constraint between C and B, on the class B is? *        2/2

1                                                                                  ✓

✓   31. How many dependencies is the A class connected to? *                    2/2

1                                                                                  ✓

✓   32. The maximum value of the multiplicity constraint between E and B, on *2/2 the B class is?

10                                                   ✓

Este formulario se creó en espe.edu.ec. Denunciar abuso

Google Formularios