Examen 1

CI3641 – Lenguajes de Programación I
Enero–Marzo 2022
Estudiante: Gregory Muñoz
Carnet: 16-11313

Constantes:
X = 3
Y = 1
Z = 3

Enlace de github donde se encontrarán los programas:
https://github.com/mvgregoryj/CI3641-Lenguajes-de-Programacion-I/tree/main/Examen%201

Respuesta 1:

Lenguaje escogido: Go

a.i) -Tipo de Alcances:

Go tiene un **alcance estático/léxico** usando bloques.

El alcance de un identificador pre-declarado es el bloque universo.
El alcance de un identificador que denota una constante, tipo, variable o función (pero no método) declarada en el nivel superior (fuera de cualquier función) es el bloque paquete.
El alcance del nombre de paquete de un paquete importado es el bloque de archivo del archivo que contiene la declaración de importación.
El alcance de un identificador que denota un receptor de método, parámetro de función o variable de resultado es el cuerpo de la función.
El alcance de un identificador de constante o variable declarado dentro de una función comienza al final del ConstSpec o VarSpec (ShortVarDecl para declaraciones de variables cortas) y termina al final del bloque contenedor más interno.
El alcance de un identificador de tipo declarado dentro de una función comienza en el identificador en el aspecto tipo y termina al final del bloque contenedor más interno.
Un identificador declarado en un bloque puede volver a declararse en un bloque interior. Si bien el identificador de la declaración interna está en el ámbito de aplicación, indica la entidad declarada por la declaración interna.

-Tipo de Asociaciones:
No hay información sobre el tipo de asociaciones que posee.

a.ii) -Tipos de módulos (si tiene):

-Diferentes formas de importar y exportar nombres

Al importar más de un paquete, se puede crear un bloque.

Forma 1:

bloque de importación de ejemplo antes de la aplicación de formato:

```
import (
  "fmt"
  "os"
  "github.com/digital/ocean/godo"
  "github.com/sammy/foo"
  "math/rand"
  "github.com/sammy/bar"

)
```

Forma 2:

```
package main

import (
 f "fmt"
  "math/rand"
)

func main() {

  for i := 0; i < 10; i++ {

    f.Printf("%d) %d\n", i, rand.Intn(25))

  }

}
```

Forma 2:

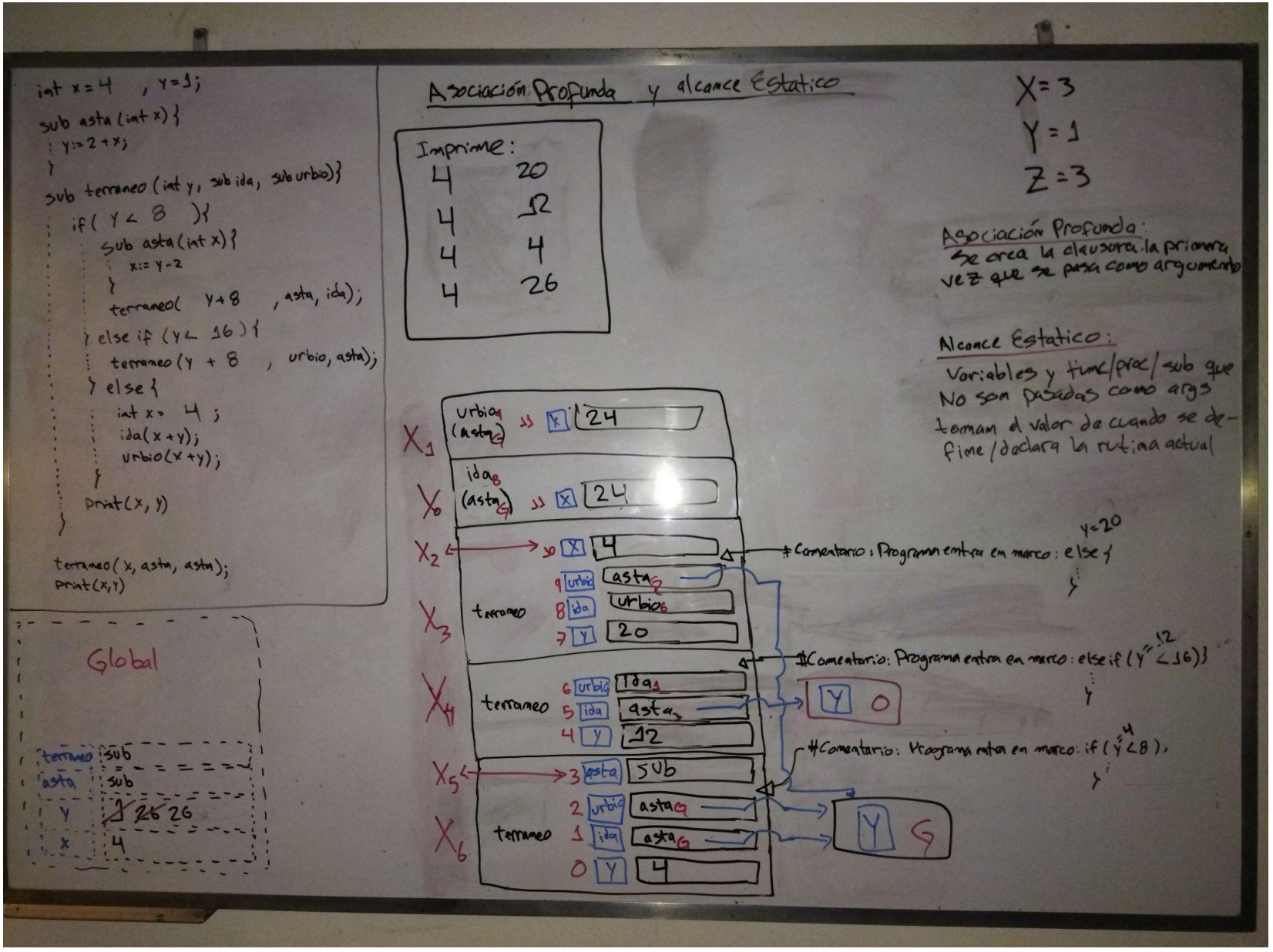
a.iii) -¿Ofrece la posibilidad de crear alises?
No.

-¿Ofrece la posibilidad de crear sobrecarga?
No, Go no tiene funciones sobrecargadas, la función más útil de sobrecarga es la de llamar a una función con argumentos opcionales e inferir valores predeterminados para los omitidos, se puede simular usando una función variable.

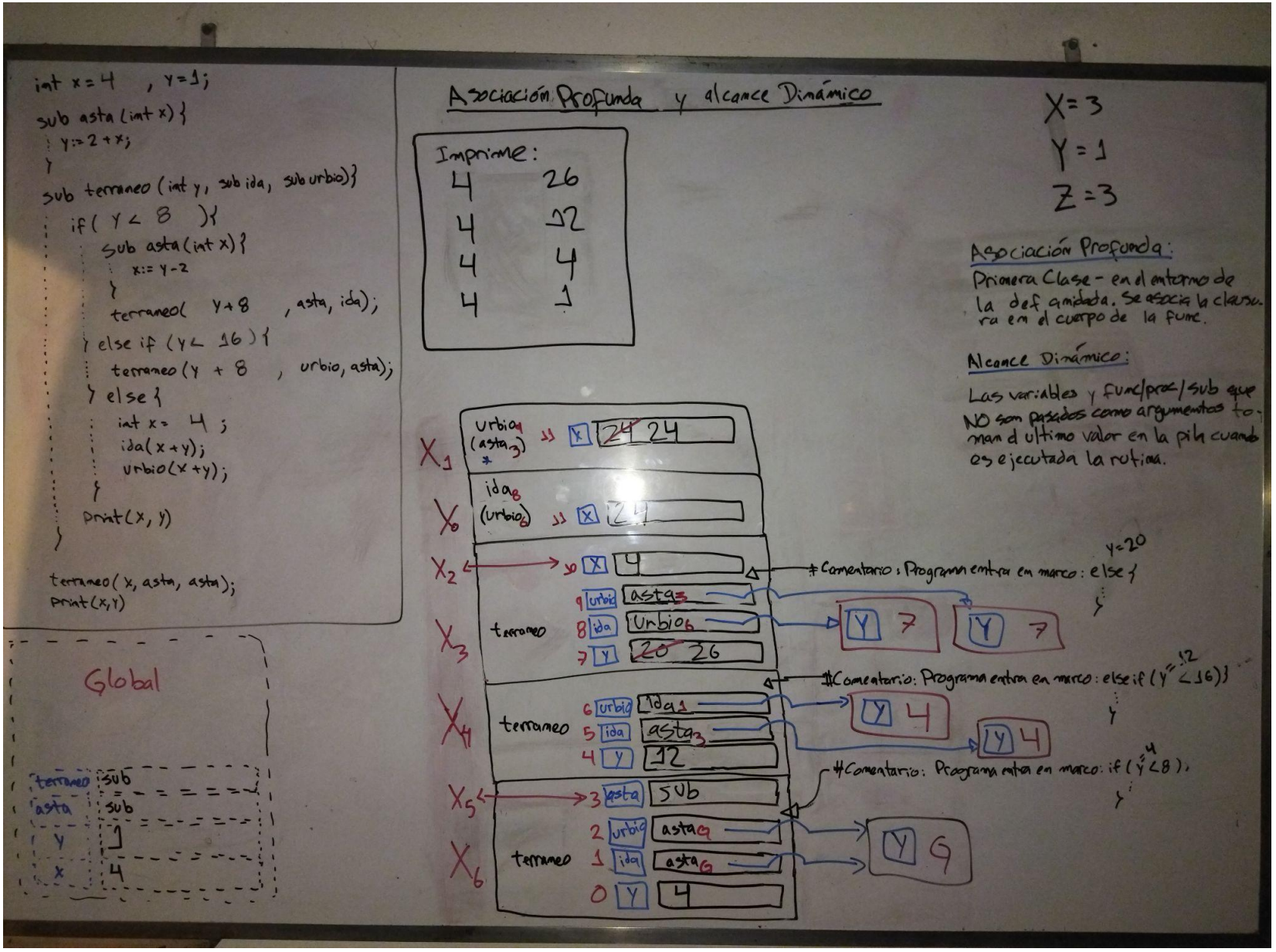
-¿Ofrece la posibilidad de crear polimorfismo?
Sí, Golang es un lenguaje orientado a objetos ligero y admite polimorfismo solo a través de interfaces.
- Dar ejemplos de alises, sobrecarga, polimorfismo

a.iv) - Qué herramientas ofrece a potenciales desarrolladores, como:
compiladores, intérpretes, debuggers, profilers, frameworks, etc.
debuggers: Delve
Frameworks: Gin · Beego · Iris · Echo · Fiber

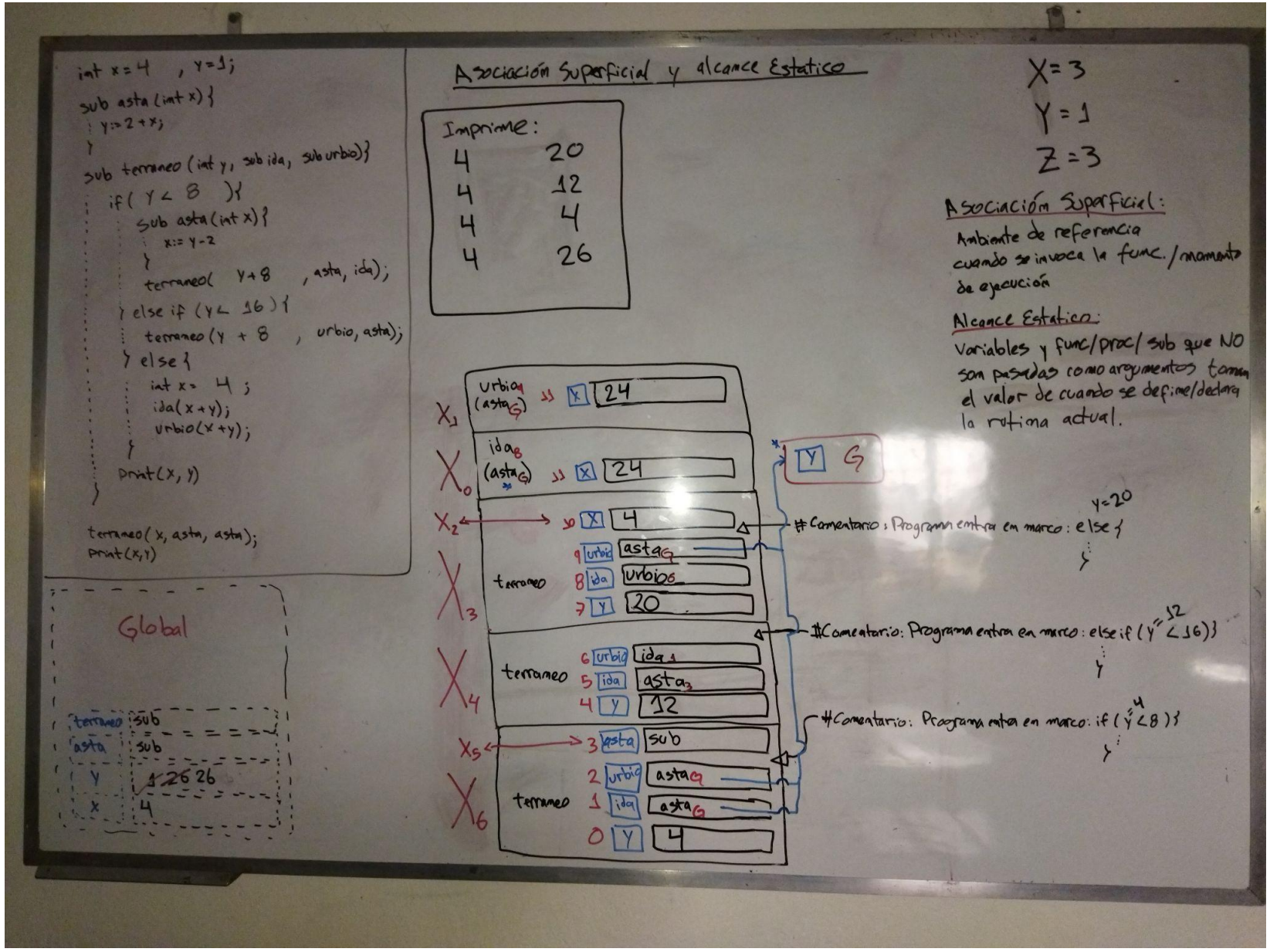
PREGUNTA 2: (a) Alcance estático y asociación profunda



(b) Alcance dinámico y asociación profunda



(c) Alcance estático y asociación superficial



(d) Alcance dinámico y asociación superficial