## Problem Statement 3: Explain the code snippet

Explain what the following code is attempting to do? You can explain by:

1. Explaining how the highlighted constructs work?
2. Giving use-cases of what these constructs could be used for.
3. What is the significance of the for loop with 4 iterations?
4. What is the significance of make(chan func(), 10)?
5. Why is “HERE1” not getting printed?

|  |
| --- |
| package main  import "fmt"  func main() {  **cnp := make(chan func(), 10)**  **for i := 0; i < 4; i++ {**  **go func() {**  **for f := range cnp {**  **f()**  **}**  **}()**  **}**  **cnp <- func() {**  **fmt.Println("HERE1")**  **}**  fmt.Println("Hello")  } |

**Q: Explaining how the highlighted constructs work?**

**Answer:** **cnp := make(chan func(), 10) :** declaring a buffered channel of type func() limited to 10 items that is going to take 10 func() values.

**for i := 0; i < 4; i++… :** a for loop of four iterations, will start four goroutines and each goroutine iterates on cnp channel and calling all items functions are there in cnp (there was no item though).

**cnp <- func()… :** assigning a value of type func() which is printing “HERE1” with ending new line character.

**Q: Giving use-cases of what these constructs could be used for.**

**Answer: make(chan func, 10) :** when there is a need of communication between two concurrently executing functions with fixed number of messages (values) to be shared, buffered channel are used.

**for i := 0; i < 4; i++…:** for loop is used to iterate over slices, arrays, maps, and channels. When there is a need to process iterable object asynchronously, goroutines are started to perform tasks, it boosts performance of program e.g. counting words occurrence in a text file, a goroutine can be started to process each line.

|  |
| --- |
| package main  import "bufio"  func main() {  file := …  sc := bufio.NewScanner(file)  for sc.Scan() {  line := sc.Text()  go func() {  // process line  }()  }  } |

**cnp <- func()… :** “<-” is used to send value on channel so that receiver can take and process it.

**Q: What is the significance of the for loop with 4 iterations?**

**Answer:** In 4 iterations this for loop starts a goroutine on each iteration.

**Q: What is the significance of make(chan func(), 10)?**

**Answer:** it constructs a buffered channel of size 10. buffered channel limits number of values without corresponding receiver. No value can sent when buffer is full and can’t be received when buffer is empty.

**Q: Why is “HERE1” not getting printed?**

**Answer:** because func value is not passed before the channel is being iterated in goroutine and also there is no receiver like select statement to wait for value, program is finishing just after passing value and printing a Hello.