LESSON 9

main.rs

```rust
let top: Vec<_> = scores.iter()
  .filter(|s| **s > 90)
  .collect();
```

Rust

Vec · HashMap · iter

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUST PROGRAMMING

LESSON 9

Collections, Closures and Iterators

Vec, HashMap, closures, iterator adapters and collect

Vec

main.rs

```rust
let mut scores = vec![90, 85, 77];
scores.push(92);
scores.len();                    // 4
scores[0];                       // 90
scores.get(10);                  // None, no panic
scores.sort();
scores.retain(|&s| s >= 80);     // keep 85, 90, 92
```

- Vec is a growable array on the heap. The vec macro creates one with values. push adds to …
- Borrow a Vec as a slice when a function only needs to read it.

01

HashMap

main.rs

```rust
use std::collections::HashMap;

let mut counts: HashMap<&str, u32> = HashMap::new();
for w in text.split_whitespace() {
    *counts.entry(w).or_insert(0) += 1;
}
if let Some(n) = counts.get("kabul") { println!("{n}"); }
```

- HashMap stores key-value pairs. The entry API makes counting and grouping one line: …
- Iteration order is not defined. Use BTreeMap when you need sorted keys.

02

Closures

main.rs

```
let bonus = 5;
let add_bonus = |s: i32| s + bonus;       // captures bonus
add_bonus(90);                            // 95

let mut count = 0;
let mut inc = || count += 1;              // FnMut
inc(); inc();

let owned = move || println!("{bonus}");
```

- A closure is an anonymous function written with pipes around its parameters. It can capture …
- move forces the closure to take ownership of what it captures, which you need for threads.

03

Iterator Adapters



- scores.iter() — iterator
- filter — |s| **s >= 80
- map — |s| s * 2

04

Putting It Together

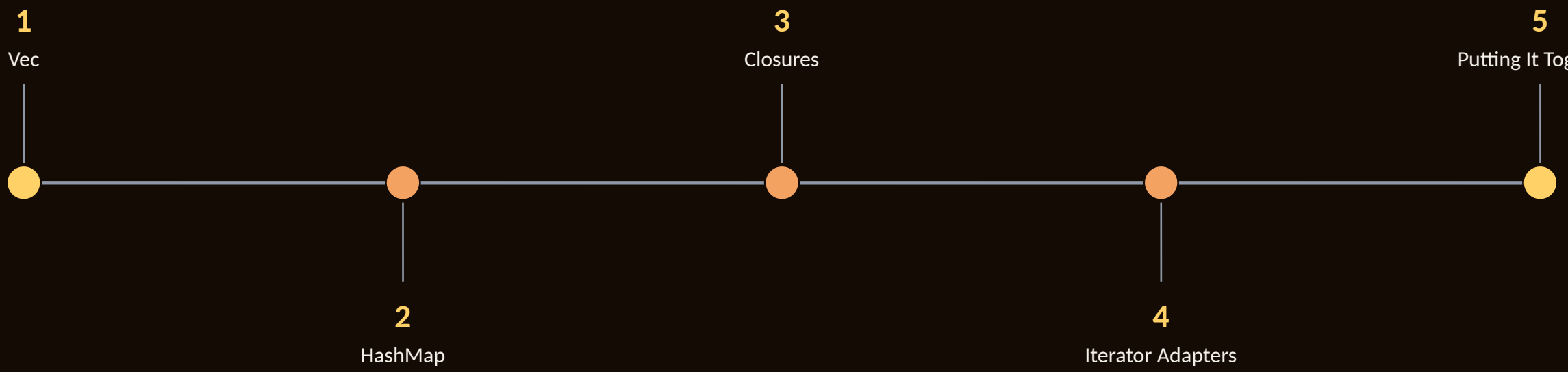
main.rs

```rust
let mut by_city: BTreeMap<&str, Vec<f64>> = BTreeMap::new();
for s in students.iter().filter(|s| s.score >= 50.0) {
    by_city.entry(&s.city).or_default().push(s.score);
}
for (city, v) in &by_city {
    let avg = v.iter().sum::<f64>() / v.len() as f64;
    println!("{city:<10} {avg:.1}");
}
```

- Here we read student records, keep those who passed, group average scores by city, and …
- into_iter consumes a collection, iter borrows it, and iter_mut borrows it mutably.

05

Lesson Map


1
Vec
2
HashMap
3
Closures
4
Iterator Adapters
5
Putting It Together

Sources and Further Reading

main.rs

```rust
let mut scores = vec![90, 85, 77];
scores.push(92);
scores.len();                    // 4
scores[0];                       // 90
scores.get(10);                  // None, no panic
scores.sort();
scores.retain(|&s| s >= 80);     // keep 85, 90, 92
```

RUST PROGRAMMING · 9

main.rs

```rust
let bonus = 5;
let add_bonus = |s: i32| s + bonus;       // captures bonus
add_bonus(90);                            // 95

let mut count = 0;
let mut inc = || count += 1;              // FnMut
inc(); inc();

let owned = move || println!("{bonus}");
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari