



First Encounter: Running Safe Code: Code Companion

Reference code for the First Encounter: Running Safe Code lecture. Sections correspond to the lecture document.

Section 1: The Anatomy of a Rust Entry Point

```
fn main() {  
    // The simplest entry point: no arguments, returns unit type ()  
    // Rust's runtime has already initialized safely before this runs  
  
    println!("=== Buffer Overflow Prevention in Rust ===\n");  
  
    // ... demonstration code follows  
}
```

The implicit return type is `()` (unit). For error-handling entry points, you could use `fn main() -> Result<(), E>` or `fn main() -> ExitCode`.

Section 2: Module Imports and the Crate Ecosystem

```
// Import the buffer_overflow module from our crate  
use rust_memory_safety_examples::buffer_overflow;  
  
fn main() {  
    // Call functions using the module path  
    buffer_overflow::safe_array_access();  
    //      ^-- module      ^-- function  
  
    buffer_overflow::safe_string_handling();  
  
    buffer_overflow::compare_c_vs_rust();  
}
```


