Rust Programming Cheat Sheet

Rust is a programming language focusing on speed, concurrency, and safety.

Includes

```
[package]
name = "example"
version = "0.1.0"
edition = "2021"
[dependencies]
rand = "0.8.4"
```

List library dependencies in the file **`cargo.toml`**

```
use std::io;
use std::cmp::Ordering;
use rand::Rng;
```

List dependencies in your code

Functions

```
fn celsius(fahr: i32) -> i32 {
  let cel = (fahr - 32) * 5 / 9;
  return cel; }
```

`fn` indicates a function

`->` designates the return type

```
fn main() {
  let cel = celcius(10);
  println!("{}", cel); }
```

Default return type (when no **`return`** statement is used) is **`()`**

Operators

`a & b`	Bitwise AND (1 if both bits are 1)	`<`	`>`	greater/less than
`a | b`	Bitwise OR (1 if either bits are 1)	`<=`	`>=`	greater/less equal
`a ^ b`	Bitwise XOR (1 if bits differ)	`==`	`!=`	equal / not equal
`a<<n`	Shift bits to the left	`&&`	`||`	and / or
`a>>n`	Shift bits to the right	`!`		not

Rust Programming Cheat Sheet

Variables	
`let myvar = "bar";`	Set immutable variable **myvar** containing the string "bar"
`let myvar = 100;`	Set immutable variable **myvar** containing the integer 100
`let myvar: i32 = 100;`	Set immutable variable **myvar** as a 32-bit integer of 100
`let myvar: i64;`	Create variable **myvar** as a 64-bit integer
`let mut myvar: i32;`	Create mutable variable **myvar** as 32-bit integer
`myvar = myvar+1;`	Increment myvar `myvar += 1;`
`let n = 100 as char;`	Create immutable variable **n**, cast as a char

If / else

```
if n < 0 { //statement; }
else if n == 0 { //statement; }
else { //statement; }
```

Loop

```
let mut count: i32 = 0;
loop { count += 1;
  if count == 10 { break; } }
```

While

```
let mut n: i32 = 0;
while n < 3 { println!("Hello");
  n = n+1; }
```

For

```
for n in 1..21 {
  println!("{}", n);
}
```

Iterator

```
let names = vec!["Alice", "Bob", "Carol"];

for name in names.iter() {
    println!("Hello {}", name);
}
```