

## Week – 9

Requires Package Creation to execute this code

Code:

```
use std::io;

fn main() {
    println!("Week-9 (A)");
    println!("Hello, World!\n");

    println!("Week-9 (B)");
    week9b();

    println!("\nWeek-9 (C)");
    week9c();

    println!("\nWeek-9 (D)");
    println!("Enter any 2 numbers: ");
    let mut input = String::new();
    io::stdin()
        .read_line(&mut input)
        .expect("Failed to read line");
    let a: i32 = input.trim().parse().expect("Invalid input");

    let mut input = String::new();
    io::stdin()
        .read_line(&mut input)
        .expect("Failed to read line");
    let b: i32 = input.trim().parse().expect("Invalid input");
    week9d(a, b);

    println!("\nWeek-9 (E)");
    week9e(a, b);

    println!("\nWeek-9 (F)");
    week9f(a, b);
}

fn week9b() {
    let i: i32 = 42;
    let f: f64 = 3.14;
    let c: char = '😍';
    let b: bool = true;

    println!("Integer: {}", i);
```

```

println!("Floating Point: {}", f);
println!("Character: {}", c);
println!("Boolean: {}", b);
}

fn week9c() {
    let name = "Harish";
    let age = 20;

    println!("My name is {} and I am {} years old.", name, age);
    println!("Formatted Age: {:05}", age);
    println!("Formatted PI: {:.2}", 3.14159);
}

fn week9d(x: i32, y: i32) {
    println!("Sum: {}", x + y);
    println!("Difference: {}", x - y);
    println!("Product: {}", x * y);
    println!("Quotient: {}", x / y);
    println!("Remainder: {}", x % y);
}

fn week9e(a: i32, b: i32) {
    println!("Bitwise AND: {}", a & b);
    println!("Bitwise OR: {}", a | b);
    println!("Bitwise XOR: {}", a ^ b);
    println!("Logical AND: {}", a > 0 && b > 0);
    println!("Logical OR: {}", a > 0 || b > 0);
    println!("Logical NOT: {}", !(a > 0));
}

fn week9f(mut x: i32, mut y: i32) {
    println!("Before Swapping, x = {}, y = {}", x, y);
    x = x + y;
    y = x - y;
    x = x - y;
    println!("After Swapping, x = {}, y = {}", x, y);
}

```

Week – 10

Code:

```
fn main() {  
    arr();  
    tup();  
}  
  
fn arr() {  
    println!("Demonstration of Arrays: ");  
    let a: [i32; 6] = [42, 57, 95, 21, 32, 85];  
    println!("Array Elements: {:?}", a);  
    println!("Accessing 5th element: ");  
    println!("5th element = {}", a[4]);  
}  
  
fn tup() {  
    println!("\nDemonstration of Tuples: ");  
    let b: (&str, i32, bool) = ("Harish", 20, true);  
    println!("Tuple Elements: {:?}", b);  
    println!("Accessing 2nd Element: ");  
    println!("2nd Element = {}", b.1);  
}
```

## Week – 11

Code:

```
fn main() {  
  // While loop  
  let mut count = 0;  
  while count < 5 {  
    println!("While loop count: {}", count);  
    count += 1;  
  }  
  
  // For loop  
  for i in 1..=5 {  
    println!("For loop count: {}", i);  
  }  
  
  // Loop with break  
  let mut i = 0;  
  loop {  
    println!("Loop count: {}", i);  
    i += 1;  
    if i >= 5 {  
      break;  
    }  
  }  
  
  // Conditional loop - while let  
  let mut optional_number = Some(5);  
  while let Some(number) = optional_number {  
    println!("Conditional loop: {}", number);  
    optional_number = None;  
  }  
}
```

Week – 12

Code:

```
fn fun1(s: String){  
    println!("This '{}' value is passed from Main Function\n",s);  
}
```

```
fn fun2(t: i64) -> i64{  
    t*t*t  
}
```

```
fn main() {  
    println!("Assigning Values to Variables");  
    let x = 5;  
    let y = x;  
    println!("Assigned Values are: x = {}, y = {}\n",x,y);  
  
    println!("Passing Values to Functions");  
    let i = String::from("Harish");  
    fun1(i);  
  
    println!("Returning Values from Functions");  
    let z = fun2(x);  
    println!("The returned Value from Function is = {}",z);  
}
```

Week – 13

Requires package creation to execute this code

Code:

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

fn main(){

    let mut input = String::new();
    io::stdin().read_line(&mut input).expect("Failed to read line");
    let x: i32 = input.trim().parse().expect("Invalid input");

    let mut input = String::new();
    io::stdin().read_line(&mut input).expect("Failed to read line");
    let y: i32 = input.trim().parse().expect("Invalid input");

    let r = rand::thread_rng().gen_range(x..=y);
    println!("Random Number generated between {} and {} is = {}",x,y,r);
}
```

Week – 14

Requires Package creation to execute this code

Code:

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

fn main() {
    let secret_number = rand::thread_rng().gen_range(1..101);

    println!("Guess the secret number between 1 and 100");

    loop {
        let mut guess = String::new();

        io::stdin().read_line(&mut guess).expect("Failed to read line");

        let guess: i32 = match guess.trim().parse() {
            Ok(num) => num,
            Err(_) => continue,
        };

        if guess == secret_number {
            println!("Congratulations, you guessed right!");
            break;
        } else if guess < secret_number {
            println!("Too low! Try again.");
        } else {
            println!("Too high! Try again.");
        }
    }
}
```