

# Presidential Initiative for Artificial Intelligence and Computing (PIAIC)

## PIAIC Batch 4-35 IoT

<https://www.piaic.org>

Internet of Things (IoT) Specialist Program

Quarter 2: Rust Programming

### Assignment # 2

1. Define Traits in your own words?

**ANSWER:**

Traits are a way to group method signatures together to define a set of behaviours necessary to accomplish some purpose.

2. Illustrate with an example how using Traits can help you organize your code and reduce duplication in your program

**ANSWER:**

For user defined data types like Car, Aeroplane, Train can implement a trait called speed which is common in all three data types.

3. Write a rust program:
  - a. Define a struct IOT\_student with attributes (name, age, education).
  - b. Define another struct IOT\_instructor (name, age).
  - c. Define a trait Questions with method ask\_Questions with a default implementation which prints ("Zoom session will be LIVE, Zoom recording will not be available. Quarter 2 studio recorded videos are available on Portal.").
  - d. Impl trait Questions for IOT\_instructor which overrides the default implementation of method ask\_question, takes student name as a parameter and prints on screen ("{} In case of any issue email to education@piaic.org").
  - e. Create instances of both the structs and call Method ask\_question.

**ANSWER:**

```

1  //a. Define a struct IOT_student with attributes (name, age, education).
2  #[derive(Debug)]
3  struct IOT_student {
4      name: String,
5      age: i32,
6      education: String
7  }
8
9  //b. Define another struct IOT_instructor (name, age).
10 #[derive(Debug)]
11 struct IOT_instructor {
12     name: String,
13     age: i32
14 }
15
16 //c. Define a trait Questions with method ask_Questions with a default
17 //implementation which prints ("Zoom session will be LIVE, Zoom recording will
18 // not be available. Quarter 2 studio recorded videos are available on Portal.").
19 trait Questions {
20     fn ask_questions(&self, name: String) {
21         println!("Zoom session will be LIVE, Zoom recording will not be available.
22             Quarter 2 studio recorded videos are available on Portal.")
23     }
24 }
25
26 //d. Impl trait Questions for IOT_instructor which overrides the default implementation
27 //of method ask_question, takes student name as a parameter and prints on
28 //screen ("{} In case of any issue email to education@piaic.org").
29 impl Questions for IOT_instructor {
30     fn ask_questions(&self, name: String) {
31         println!("{}", name);
32     }
33 }
34
35 //e. Create instances of both the structs and call Method ask_question.
36
37 fn main() {
38     let naeem = IOT_student {
39         name: String::from("Naeem"),
40         age: 28,
41         education: String::from("Master")
42     };
43     let sir_imran = IOT_instructor {
44         name: String::from("Sir Imran"),
45         age: 35
46     };
47     sir_imran.ask_questions(naeem.name);
48 }

```

4. Go through the solution of the largest function given at the end of 10.2 in the book and rewrite the solution but this time returning the smallest item instead largest.

```
1  //4. Go through the solution of the largest function given at the end of 10.2 in the book and
2  //rewrite the solution but this time returning the smallest item instead largest.
3  fn smallest<T: PartialOrd + Copy>(list: &[T]) -> T {
4      let mut smallest = list[0];
5
6      for &item in list {
7          if item < smallest {
8              smallest = item;
9          }
10     }
11     smallest
12 }
13
14 fn main() {
15     let number_list = vec![34, 50, 25, 100, 65];
16
17     let result = smallest(&number_list);
18     println!("The smallest number is {}", result);
19
20     let char_list = vec!['y', 'm', 'a', 'q'];
21
22     let result = smallest(&char_list);
23     println!("The smallest char is {}", result);
24 }
```