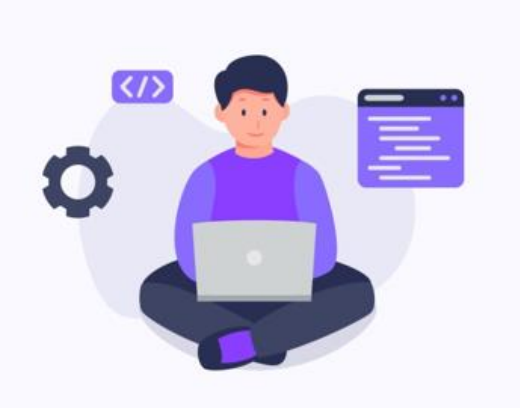

Session 6

Synchronous and Asynchronous Programming in Dart

Session Overview

- Define synchronous and asynchronous programming in Dart.
- Explain asynchronous programming with the `future` keyword.
- Explain asynchronous programming with the `async` and `await` keywords.

Synchronous Programming in Dart

Code Snippet 1:

```
import 'dart:io';

void main() {

     print("Enter your birth place :");

  String birthplace = stdin.readLineSync();

 print("Your birthplace is ${birthplace}");


}
```

Output for Code Snippet 1:

```
Enter your birth place :
New York
Your birthplace is New York

Process finished with exit code 0
```

Asynchronous Programming in Dart [1-5]

Code Snippet 2:

```
import 'dart:io';
import 'dart:async';

void main() {
 File file = new
File(Directory.current.path+"\\names.txt");
 Future<String> f = file.readAsString();

 f.then((data)=>print(data));
 print("main ends here");
```

Output for Code Snippet 2:

```
main ends here
Jon
Dan
Ron
Ricky

Process finished with exit code 0
```

Asynchronous Programming in Dart [2-5]

Code Snippet 3:

```
import 'dart:async';

void main(){

var myfutureval = Future.value(14);
print(myfutureval);

}
```

Output for Code Snippet 3:

```
Instance of 'Future<int>'

Process finished with exit code 0
```

Asynchronous Programming in Dart [3-5]

Code Snippet 4:

```
import 'dart:async';
void main(){

Future<int> getFuture() {
    return Future.error("This is an
error");
  }
 getFuture();
}
```

Output for Code Snippet 4:

```
Unhandled exception:
This is a future error

Process finished with exit code 255
```

Asynchronous Programming in Dart [4-5]

Code Snippet 5:

```
import 'dart:async';
void main(){
Future.delayed(Duration(milliseconds:
10000), () {
print("This is a delayed future");
});
}
```

Output for Code Snippet 5:

```
This is a delayed future

Process finished with exit code 0
```

Asynchronous Programming in Dart [5-5]

Code Snippet 6:

```
import 'dart:async';
void main()async{
 demo() async {
  print("Good Morning");
  }
  await demo();
  print("Have a great day!");
}
```

Output for Code Snippet 6:

```
C:/src/flutter/bin/cache/dart-sdk/bin
Good Morning
Have a great day!

Process finished with exit code 0
```

TODO Problems Terminal Dart Analysis

Summary [1-2]

- A synchronous process waits for an event to be completed before it starts to execute another event.
- An advantage of asynchronous over synchronous is that the execution of a process does not depend on any other process.
- Program loading time is slower in synchronous processes.
- Asynchronous programming fixes the chain of events in a programming cycle.
- The `async` keyword is used when declaring a function as asynchronous.
- The `await` keyword holds the currently running function until the result is ready.

Summary [2-2]

- Future value creates a future with a value. If the value is a future, the created future waits for the value to execute and then, it is executed with the same result.
- Future error creates a future, which completes with an error.
- Future delay always works with a certain amount of duration. Computation will be executed after a given duration has passed.