LAMBDA

Function as a Value

```
let f1 = function() {
  console.log("Entering Lambda1");
  console.log("Leaving Lambda1");
};

f1();

let f2 = () => {
  console.log("Entering Lambda2");
  console.log("Leaving Lambda2");
};

f2();
```

Client - Server Setup

```
nsole.log("Welcome to Higher Order Functions");

function Server()
{
  console.log("Task 1");
  console.log("Task 2");
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server();
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server();
  console.log("Leaving Client2");
}

Client1();
Client2();
```

Customising Server (Classical Style)

```
nsole.log("Welcome to Higher Order Functions");

function Server(v)
{
  console.log("Task 1");
  switch (v) {
    case 0: console.log("Task 2 - v1");
            break;
    case 1: console.log("Task 2 - v2");
            break;
    default: console.log("Task 2");

  }
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server(0);
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server(1);
  console.log("Leaving Client2");
}

function Client3()
{
  console.log("Entering Client3");
  Server();
  console.log("Leaving Client3");
}

Client1();
Client2();
Client3();
```

Callback Pattern - Higher Order Function

```
console.log("Welcome to Higher Order Functions");

function v1()
{
  console.log("Task 2 - v1");
}

function v2()
{
  console.log("Task 2 - v2");
}

function v()
{
  console.log("Task 2");
}

function Server(f = v)
{
  console.log("Task 1");
  f();
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server(v1);
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server(v2);
  console.log("Leaving Client2");
}

function Client3()
{
  console.log("Entering Client3");
```

```
console.log("Entering Client3");
Server();
console.log("Leaving Client3");
}

Client1();
Client2();
Client3();
```

Callback Pattern with Lambda (Variant 1)

```
console.log("Welcome to Higher Order Functions");

function Server(f)
{
  console.log("Task 1");
  f();
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server(
          function() {
            console.log("Task 2 - v1");
          }
        );
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server(
          function() {
            console.log("Task 2 - v2");
          }
        );
  console.log("Leaving Client2");
}

function Client3()
{
```

```
  console.log("Entering Client3");
  Server(
          function() {
            console.log("Task 2");
          }
        );
  console.log("Leaving Client3");
}

Client1();
Client2();
Client3();
```

Callback Pattern with Lambda (Variant 2)

```
nsole.log("Welcome to Higher Order Functions");

function Server(f)
{
  console.log("Task 1");
  f();
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server(
    () => console.log("Task 2 - v1")
  );
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server(
    () => console.log("Task 2 - v2")
  );
  console.log("Leaving Client2");
}

function Client3()
{
  console.log("Entering Client3");
  Server(
    () => console.log("Task 2")
  );
  console.log("Leaving Client3");
}

Client1();
Client2();
Client3();
```

Callback Pattern with Lambda (Variant 3)

```
sole.log("Welcome to Higher Order Functions");

function Server(f)
{
  const g = 55;
  console.log("Task 1");
  f(55);
  console.log("Task 3");
}

function Client1()
{
  console.log("Entering Client1");
  Server(
    v => {
      console.log("Task 2 - v1");
      console.log(`Value = ${v}`);
    }
  );
  console.log("Leaving Client1");
}

function Client2()
{
  console.log("Entering Client2");
  Server(
    v => {
      console.log("Task 2 - v2");
      console.log(`Value = ${v}`);
    }
  );
  console.log("Leaving Client2");
}

function Client3()
{
  const x = 22;
  console.log("Entering Client3");
  Server(
    v => {
      console.log("Task 2");
      console.log(`Value = ${v + x}`);
    }
  );
```

```
};
console.log("Leaving Client3");
}

Client1();
Client2();
Client3();
```

Add : 25, Patel Shopping Center, Sai Nath Road, Malad (west) ,Opp malad subway , Mumbai 64 Contact : 9820396074, 022-28809398, 9820860292