NAME : MAKVANA NEEL
ROLL NO : CE064
ID : 20CEUOS086
BATCH : A4

LAB03:

Github link:

https://github.com/NK16082002/SDP-LABS/tree/main/LAB3

Tutorial 1:

```
import 'dart:convert';
import 'dart:ffi';

import 'package:lab3_tutorial1/lab3_tutorial1.dart' as lab3_tutorial1;
import 'dart:math';

void main(List<String> arguments) {
/*
  var a = 10.5;
  a = 5;
  // (var )a = 18.5;// Error: A value of type 'double' can't be assigned to a
variable of type 'int'.
  // (num, var) a = "hi";//Error: A value of type 'String' can't be assigned to a
variable of type 'num'.
  // (var) a = "hi";//Error: A value of type 'String' can't be assigned to a
variable of type 'int'.


  print(a);


  var b;
  b=12.5;
  // b = 8;
  // b = "hii";
  // b = a;
  print(b);

  a = b;
  print(a);
*/

/*
  // infinite loop
  while(true);
  while(true){};
*/

/*
  //  Braces defines scope
  var sum = 1;
  while (sum < 10) {
    sum += 4;//1 5 9 13
    print(sum);
  }

  // If we want exicute single line we can do like this.
  var i=1;
  while(i*2>0)
    i*=2;
  print("${i} of bit ${i.bitLength}");

*/

/*
  // do while loop executes block at least one time.
  do {
    // Here it is like normal execution. Because in condition false is there
  } while (false);

  var sum = 1;
  do{
    sum += 4;
    print(sum);
  }while(sum<10);

  // differce
  sum = 11;
  while (sum < 10) {
    sum += 4;//this line never executes if sum>=10
  }

  sum = 11;
  do{
    sum += 4;//this line executes only one time if sum>=10
  }while(sum<10);

  sum = 1;
  while (true) {
    sum += 3;
    if (sum > 9) {
      break;// exit from while loop
      print("object");// not executes ever
    }
  }
*/

/*

  final random = Random(); // random is instance of '_Random'
  num x = random.nextInt(6);
  while (x + 1 != 6) {
    print(x);
    print('Not a six!');
    ///nextInt is a method that generates a random integer between 0 and one less
than the maximum value you give it( here 6).
    x = random.nextInt(6);
  }
  print('Finally, got a six!!');

*/

/*
for (var i = 0; i < 5; i++) {
  print(i);
}
for (var i = 0; i < 6; i++) {
  if(i == 2) {
    // without printing 2 go to next itteration
    continue;
  }
  if(i==4){
```

```dart
    // not prints 4 5
    break;
  }
  print(i);
}

const myString = "I ❤ Dart";
for(var codePoint in myString.runes){
  print(String.fromCharCode(codePoint));
}

const myNumbers = [1,2,3,4];
// Same things in different ways.
myNumbers.forEach((number) => print(number*3));
myNumbers.forEach((number) =>{
  print(number*3)
});

// myNumbers.forEach( number => print(number*3));//Error
*/

/*
// Mini-exercises

// 1. Create a variable named counter and set it equal to 0.
// Create a while loop with the condition counter < 10.
// The loop body should print out “counter is X” (where X
// is replaced with the value of counter) and then
// increment counter by 1.
int counter = 0;
while(counter<10){
  print("counter is ${counter}");
  counter++;
}

// 2. Write a for loop starting at 1 and ending with 10
// inclusive. Print the square of each number.
for(int i=1; i<=10; i++){
  print(i*i);
}

// 3. Write a for-in loop to iterate over the following
// collection of numbers. Print the square root of each
// number.
const numbers = [1, 2, 4, 7];
for(int number in numbers){
  print(sqrt(number));
}

// 4. Repeat Mini-exercise 3 using a forEach loop.
numbers.forEach((number)=>(print(sqrt(number))));
```

```
*/

// Challenges

// Challenge 1: Find the error
// What’s wrong with the following code?

// const ﬁrstName = 'Bob';//<- non-ascii code (ﬁ) it should be fi
// if(ﬁrstName == 'Bob') {
// const lastName = 'Smith';
// } else if (ﬁrstName == 'Ray') {
// const lastName = 'Wenderlich';
// }
//can't be used in identifiers, only in strings and comments.

// Challenge 2: Boolean challenge
// In each of the following statements, what is the value of the
// Boolean expression?
  true && true; //-> true
  false || false; //->false
  (true && 1 != 2) || (4 > 3 && 100 < 1); //true
  ((10 / 2) > 3) && ((10 % 2) == 0); //true

// Challenge 3: Next power of two
// Given a number, determine the next power of two above or
// equal to that number. Powers of two are the numbers in the
// sequence of 2¹, 2², 2³, and so on. You may also recognize the
// series as 1, 2, 4, 8, 16, 32, 64...
  int res = 1, n = 32;
  while (res < n) {
    res *= 2;
  }
  print(res);

// Challenge 4: Fibonacci
// Calculate the nth Fibonacci number. The Fibonacci sequence
// starts with 1, then 1 again, and then all subsequent numbers
// in the sequence are simply the previous two values in the
// sequence added together (1, 1, 2, 3, 5, 8...). You can get a
// refresher here:
  int nth = 3, prev = 1, curr = 1;
  for (int i = 2; i < nth; i++) {
    int t = prev;
    prev = curr;
    curr = curr + t;
  }
  print(curr);

// Challenge 5: How many times?
// In the following for loop, what will be the value of sum, and
// how many iterations will happen?
  var sum = 0;
  for (var i = 0; i <= 5; i++) {
    sum += i;
  }
//-> 6 times

// Challenge 6: The final countdown Print a countdown from 10 to 0.
  int count = 10;
  do {
    print(count);
    count--;
  } while (count >= 0);

// Challenge 7: Print a sequence
// Print the sequence 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
// 0.7, 0.8, 0.9, 1.0.
  for (double i = 0.0; i <= 1;) {
    print(i);
    i = i + (0.1);
  }
}
/// ASCII 0-127
/// EXPANDED ASCII 0-255
/// uni: 2bytes = 0 to 65536
/// '-> UTF
/// UTF-8
/// UTF-16
/// UTF-3
/// surrogate pairs
/// runes


// 6 lab1
// 13 lab2
// 20 lab3





/**
ASCII(American Standard Code for Information Interchange)
Originally based on the English alphabet, ASCII encodes 128(7 bit) specified
characters into seven-bit integers as shown by the ASCII chart above.
Ninety-five of the encoded characters are printable: these include the digits 0 to
9, lowercase letters a to z, uppercase letters A to Z, and punctuation symbols.
In addition, the original ASCII specification included 33 non-printing control
codes which originated with Teletype machines;
a->97
All uppercase come before lowercase letters; for example, "Z" precedes "a"
Digits and many punctuation marks come before letters
```

```
*/

/**
Expanded ASCII
(8bit)
 */
```

Tutorial 2:

```
import 'package:lab3_tutorial2/lab3_tutorial2.dart' as lab3_tutorial2;
import 'dart:math';

String compliment(int number){
  return '$number is very nice number.';
}

// Dart is optionally-typed language.
// It is possible to omit the types from your function declaration.
compliment1(int number){
  return '$number is very nice number.';
}

void helloPersonAndPet(String person, String pet) {
  print('Hello, $person, and your furry friend, $pet!');
}

String fullName(String first, String last, [String? title]) {
  if (title != null) {
    return '$title $first $last';
  } else {
    return '$first $last';
  }
}

bool withinTolerance(int value, [int min = 0, int max = 10]) {
  return min <= value && value <= max;
}

bool withinTolerance1(int value, {int min = 0, int max = 10}) {
  return min <= value && value <= max;
}

bool withinTolerance2({required int value, int min = 0, int max = 10,}) {
  return min <= value && value <= max;
}
```

```dart
// Function having side effect
void helloWSideEffect() {
  print('Hello!');
}

String helloWOSideEffect() {
  return "Hello!";
}

// Mini-Exercises
String youAreWonderful(String name, int numberPeople){
  return "You are wonderful, $name. $numberPeople people think so.";
}

// Returning Function from Function
Function namedFunction() {
  return () { print('hello'); };
}

Function applyMultiplier(num multiplier) {
  return (num value) {
    return value * multiplier;
  };
}

Function countingFunction() {
  var counter = 0;
  final incrementCounter = () {
    counter += 1;
    return counter;
  };
  return incrementCounter;
}

// Arrow Functions
int add(int a, int b) => a + b;

// Refactoring Example-2
Function applyMultiplier1(num multiplier) {
  return (num value) => value * multiplier;
}

int repeatTask(int times, int input, Function task){
  while(times-- != 0){
    input = task(input);
  }
  return input;
}

void main(List<String> arguments) {
  /*
  // Functions
  const input = 12;
  // single parameter function
  final output = compliment(input);
  print(output);    // 12 is very nice number.
  */

  /*
  // multiple parameter function
  helloPersonAndPet('Fluffy', 'Chris');     // Hello, Fluffy, and your furry
friend, Chris!
  */

  /*
  // Making parameters optional
  print(fullName('Ray', 'Wenderlich'));                   // Ray Wenderlich
  print(fullName('Albert', 'Einstein', 'Professor'));     // Professor Albert
Einstein
  */

  /*
  // Providing default values
  print(withinTolerance(5));           // true
  print(withinTolerance(15));          // false
  print(withinTolerance(9, 7, 11));    // true
  print(withinTolerance(9, 7));        // true
  */

  /*
  // Naming parameters
  print(withinTolerance1(9, min: 7, max: 11)); // true
  print(withinTolerance1(9, min: 7, max: 11)); // true
  print(withinTolerance1(9, max: 11, min: 7)); // true

  print(withinTolerance1(5));            // true
  print(withinTolerance1(15));           // false
  print(withinTolerance1(9, min: 7));    // false
  print(withinTolerance1(15, max: 20));  // true

  // Error: Too many positional arguments: 1 expected, but 3 found.
  // print(withinTolerance1(9, 7, 11));

  // Error: Too few positional arguments: 1 required, 0 given.
  // print(withinTolerance1());
  */

  /*
  // Making named parameters required
  // Error: The named parameter 'value' is required, but there's no corresponding
argument.
```

```dart
// print(withinTolerance2());
*/

/*
// Anonymous Functions
int number = 4;
String greeting = 'hello';
bool isHungry = true;
Function multiply = (int a, int b) {
  return a * b;
};

// Error: Function expressions can't be named.
// Function myFunction = int multiply(int a, int b){
//   return a * b;
// };

print(multiply(2, 3));      // 6

final triple = applyMultiplier(3);
print(triple(6));          // 18
print(triple(14.0));       // 42.0
*/

/*
// Anonymous fuction in forEach Loop
const numbers = [1, 2, 3];
numbers.forEach((number) {
  final trippled  = number*3;
  print(trippled);
});
// 3 6 9
*/

/*
// Closure
final counter1 = countingFunction();
final counter2 = countingFunction();

print(counter1()); // 1
print(counter2()); // 1
print(counter1()); // 2
print(counter1()); // 3
print(counter2()); // 2
*/

/*
// Mini-Exercises
Function wonderful = (String name){
  return "You are wonderful, $name.";
};
```

```
  const people = ['Chris', 'Tiffani', 'Pablo'];
  people.forEach((person) { print(wonderful(person)); });
  // You are wonderful, Chris.
  // You are wonderful, Tiffani.
  // You are wonderful, Pablo.
  */

  /*
  // Arrow Functions
  final multiply = (int a, int b) => a * b;
  print(multiply(2, 3)); // 6
  */

  /*
  // Refactoring Example-3
  const numbers = [1, 2, 3];
  numbers.forEach((number) => print(number * 3));
  */

  /*
  // Mini-Exercises
  const people = ['Chris', 'Tiffani', 'Pablo'];
  people.forEach((person) => print("You are wonderful, $person."));
  */

  /*
  // Challenges
  // Challenge-1: Prime Time
  Function isPrime = (n) {
    for(int i = 2; i <= sqrt(n); i++){
      if(n%i == 0)
        return false;
    }
    return true;
  };
  print(isPrime(19));   // true
  print(isPrime(20));   // false

  // Challenge-2: Can you repeat that?
  // Function repeatTask implemented above...
  print(repeatTask(4, 2, (n) { return n*n; }));

  // Challenge-3: Dart and Arrows
  print(repeatTask(4, 2, (n) => n*n ));
  */
}
```

Tutorial 3:

```dart
import 'package:lab3_tutorial3/lab3_tutorial3.dart' as lab3_tutorial3;
// import 'package:characters/characters.dart';

void main(List<String> arguments) {
/*
  // Getting Characters
  var salutation = "Hello!";
  print(salutation.codeUnits);    // [72, 101, 108, 108, 111, 33]
*/
/*
  const dart = '🎯';
  print(dart.codeUnits);          // [55356, 57263]
  print(dart.runes);              // (127919)
*/
/*
  // Mongolian Flag
  const flag = '🇲🇳';
  print(flag.runes);              // (127474, 127475)
*/
/*
  // Family: Man, Woman, Girl, Boy
  const family = '👨‍👩‍👧‍👦';
  print(family.runes);            // (128104, 8205, 128105, 8205, 128103, 8205,
128102)
  print(family.length);           // 11
  print(family.codeUnits.length); // 11
  print(family.runes.length);     // 7
*/
}
```