NAME: MUHAMMAD MUBASHIR

FATHER NAME: SAEED AKBER

COURSE: MOBILE APPLICATION

COURSE INSTRUCTOR: SALMAN BEDIYA

Concept of Dart Programming Language

Assignment 1 Question Task

1.Check whether a given year is a leap year or not?

2.Check whether a given year is a leap year or not?

3.Check the maximum number between two numbers.

4.Check if a number is positive or negative.

5.Check whether a number is divisible by 5 and 11 or not.

Assignment 2 Question Task

6.Write a program to check if a given string is a palindrome.\

7.Write a program to calculate the factorial of a given number using a function.

8.Write a program to print out the Fibonacci sequence up to a given number.

9.Write a program to calculate the distance between two points on a 2D plane using a function.

10. Write a program to convert a temperature from Fahrenheit to Celsius using a function.

11.Write a program to calculate the area of a circle using a function.

12.Write a program to print out the prime numbers between 1 and a given number.

TASK CREATING CLASS CONCEPT

13.Create a class called "BankAccount" with the following attributes:

account_number (integer)

balance (double)

account_type (string)

interest_rate (double)

And the following methods:

deposit(amount): adds the amount to the balance.

withdraw(amount): subtracts the amount from the balance. You cannot withdraw more than the current balance.

add_interest(): adds interest to the balance based on the interest rate.

display(): prints out the account number, balance, account type, and interest rate.

Then, create two instances of the BankAccount class, each with its own account number, balance, account type, and interest rate.

Finally, call the deposit(), withdraw(), add_interest(), and display() methods on each instance and confirm that the information is updated and displayed correctly.

14.(a)Create a class called "Student" with the following attributes:

name (string)

id (string)

courses (list of strings)

And the following methods:

add_course(course): adds a course to the student's list of courses.

drop_course(course): removes a course from the student's list of courses.

display_courses(): prints out the student's list of courses.

Then, create two instances of the Student class, each with their name, id, and courses.

Finally, call the add_course(), drop_course(), and display_courses() methods on each instance and confirm that the information is updated and displayed correctly.

Password Generate Task

15.(a)Write a program that generates a random password based on the user's specifications. The user should be able to specify the length of the password and whether it should include numbers, letters, and special characters.

(b)Write a function that takes a string input from the user and checks if the password is strong enough based on certain criteria (e.g. minimum length, use of uppercase letters, use of special characters, etc.).

If Else Condition

16.Write a program that takes an integer input from the user and prints out numbers from 1 to that integer, but for multiples of 3 print "Fizz" instead of the number, and for multiples of 5 print "Buzz". For numbers that are multiples of both 3 and 5, print "FizzBuzz".

17.Write a function that takes a list of numbers as input and sorts the list in ascending order.

TASK INHERITANCE

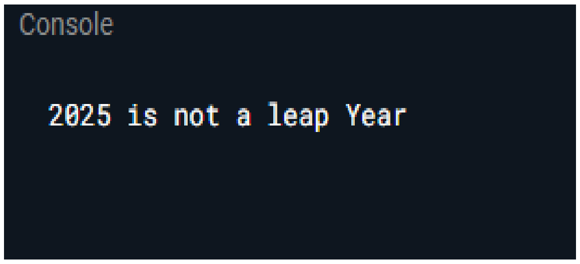
18.Calculate area of different shapes. Create a base class named “Shape” with a method to return the CalculateArea of the shape. Create a class named “Rectangle”, “Circle” and “Square” derived from the base class “Shape”.

TASK 1 to 5

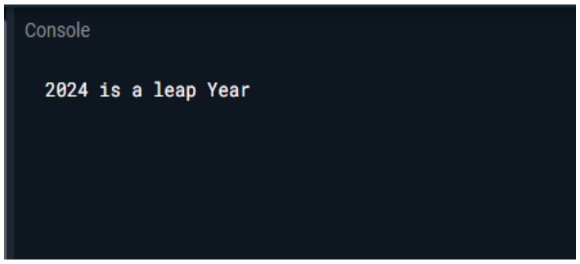
<table>
<tr><th>TASK 1</th><th>OUTPUT</th></tr>
<tr><td>

```
//Check whether a given year is a leap year or
not?

Void main(){
int year = 2024
// int year = 2025;

  if ((year % 4 == 0 && year % 100 !=0)    ||
year %  400 == 0){
    print('$year is a leap Year');
   }

 else{
   print("$year is not a leap Year");
  }
}
```

</td><td>



```
Console

 2025 is not a leap Year
```



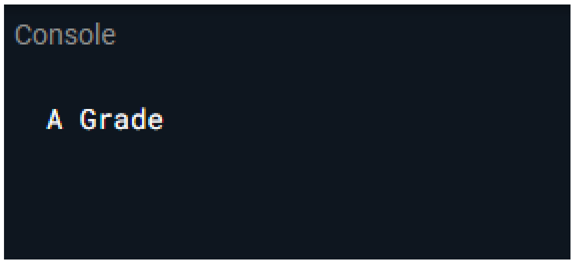
```
Console

 2024 is a leap Year
```

</td></tr>
</table>

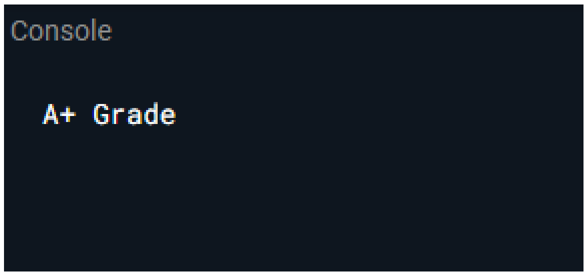
<table>
<tr><th>TASK 2</th><th>OUTPUT</th></tr>
<tr><td>

```
 // Program to check the grade of a student
from total marks with the criteria like A =
>80, B = >70, etc.

void main(){
// int a = 39;
// int a = 89;
 int a = 79;

 if(a>80 && a<100){
   print("A+ Grade");
      }

 else if(a>70 && a<80){
   print("A Grade");
     }
```

</td><td>



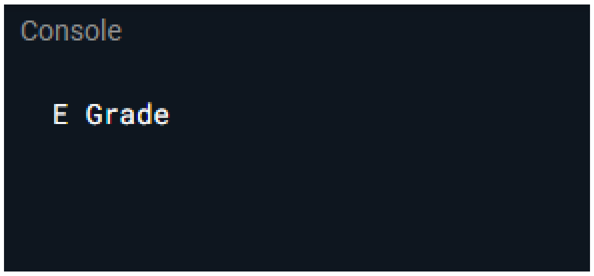
```
Console

 A Grade
```



```
Console

 A+ Grade
```

</td></tr>
</table>

```
  else if(a>60 && a<70){
    print("B Grade");
      }
  else if(a>50 && a<60){
    print("C Grade");
     }

  else if(a>40 && a<50){
    print("D Grade");
     }

  else if(a>33 && a<40){
    print("E Grade");
     }

  else if(a>0 && a<33){
    print("F Grade");
    }
else{
print("Correct the valid input");
    }
}
```



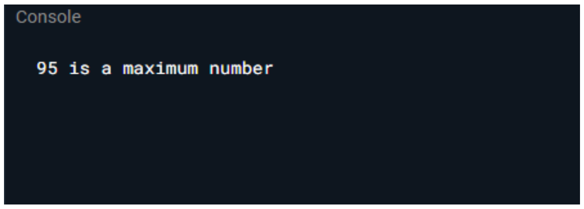
Console

```
E Grade
```

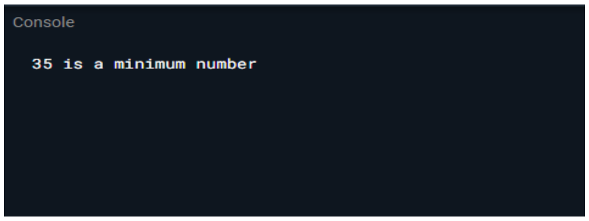
TASK 3

```
 // Check the maximum number between two
numbers

void main(){
var a, b;
  a = 95;
  b = 37;
//  a = 35;
// b = 87;

if(a>b){
    print("$a is a maximum number");
    }

  else if(a<b){
    print("$a is a minimum number");
      }
  }
```

OUTPUT



```
Console

  95 is a maximum number
```

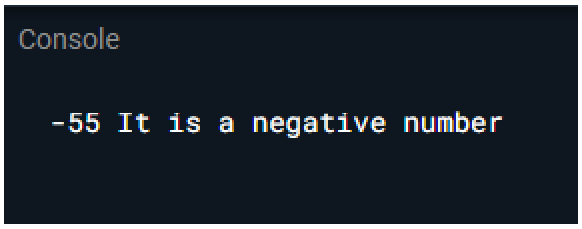


```
Console

  35 is a minimum number
```

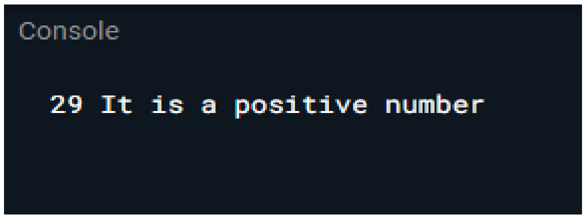
TASK 4

```
 // Check if a number is positive or negative
void main() {
  var b = 0;
  var num1 = 29;
 // var num1 = -55;
  if(num1>b){
    print("$num1 It is a positive number");
      }
 else if(num1<b){
    print("$num1 It is a negative number");
      }
}
```

OUTPUT



```
Console

  -55 It is a negative number
```

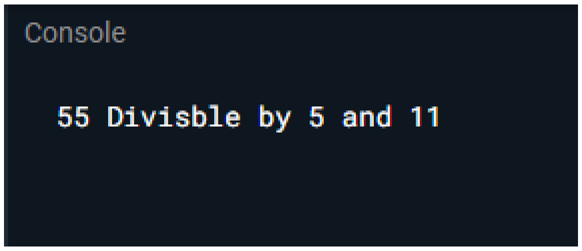


```
Console

  29 It is a positive number
```

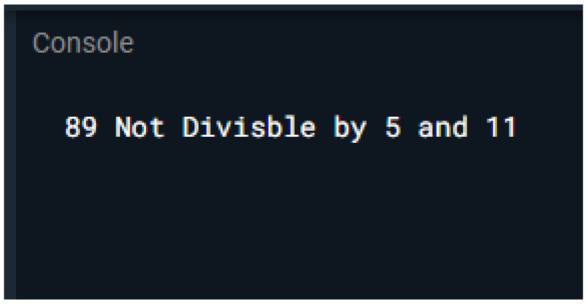
TASK 5

```
// Check whether a number is divisible by 5
and 11 or not.

void main() {
int a, b ,c;
  a=5;
  b=11;
//  c = 89;
     c = 55;

  if(c % a == 0 && c % b == 0)
  {
    print("$c Divisible by 5 and 11");
  }
   else{
    print("$c Not Divisible by 5 and 11");
  }
}
```

OUTPUT



```
55 Divisble by 5 and 11
```



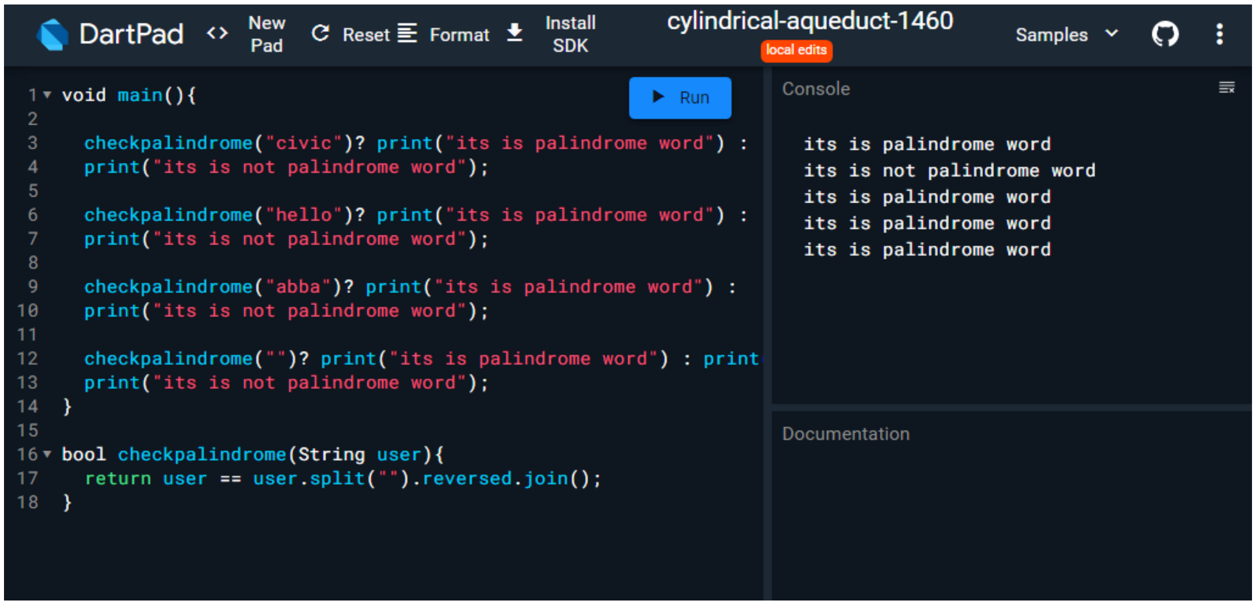
```
89 Not Divisble by 5 and 11
```

Assignment 2

1. Write a program to check if a given string is a palindrome.

```
void main(){


  checkpalindrome("civic")? print("its is palindrome word") :
  print("its is not palindrome word");


  checkpalindrome("hello")? print("its is palindrome word") :
```

```
  print("its is not palindrome word");


  checkpalindrome("abba")? print("its is palindrome word") :
  print("its is not palindrome word");


  checkpalindrome("")? print("its is palindrome word") : print("its is not
palindrome word");    checkpalindrome("amma")? print("its is palindrome word") :
  print("its is not palindrome word");
}


bool checkpalindrome(String user){
  return user == user.split("").reversed.join();
}
```

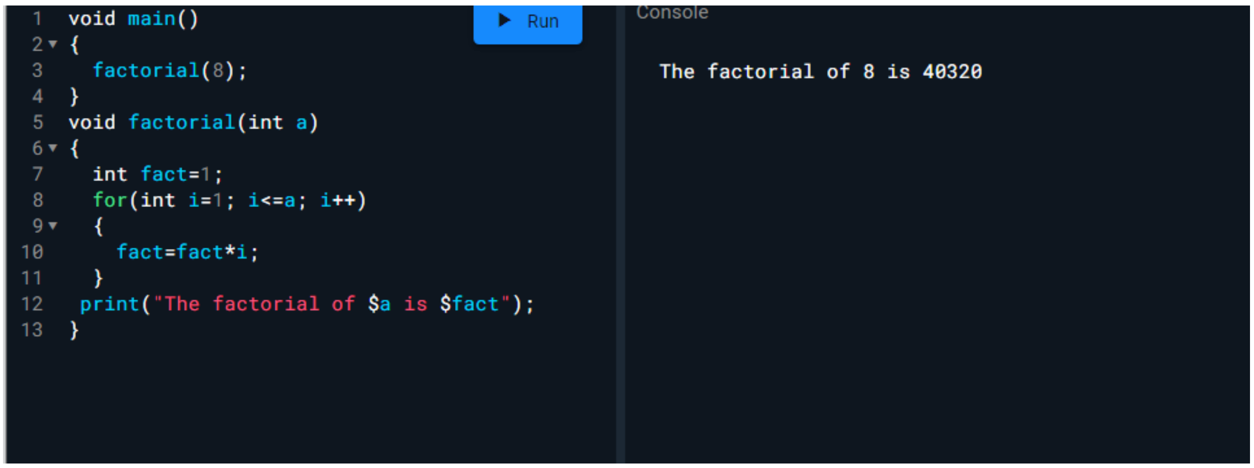


```
void main(){

  checkpalindrome("civic")? print("its is palindrome word") :
  print("its is not palindrome word");

  checkpalindrome("hello")? print("its is palindrome word") :
  print("its is not palindrome word");

  checkpalindrome("abba")? print("its is palindrome word") :
  print("its is not palindrome word");

  checkpalindrome("")? print("its is palindrome word") : print
  print("its is not palindrome word");
}

bool checkpalindrome(String user){
  return user == user.split("").reversed.join();
}
```

```
its is palindrome word
its is not palindrome word
its is palindrome word
its is palindrome word
its is palindrome word
```

2. Write a program to calculate the factorial of a given number using a function.

```
void main()
{
  factorial(8);
}
void factorial(int a)
{
  int fact=1;
  for(int i=1; i<=a; i++){
    fact=fact*i;
  }
 print("The factorial of $a is $fact");
}
```

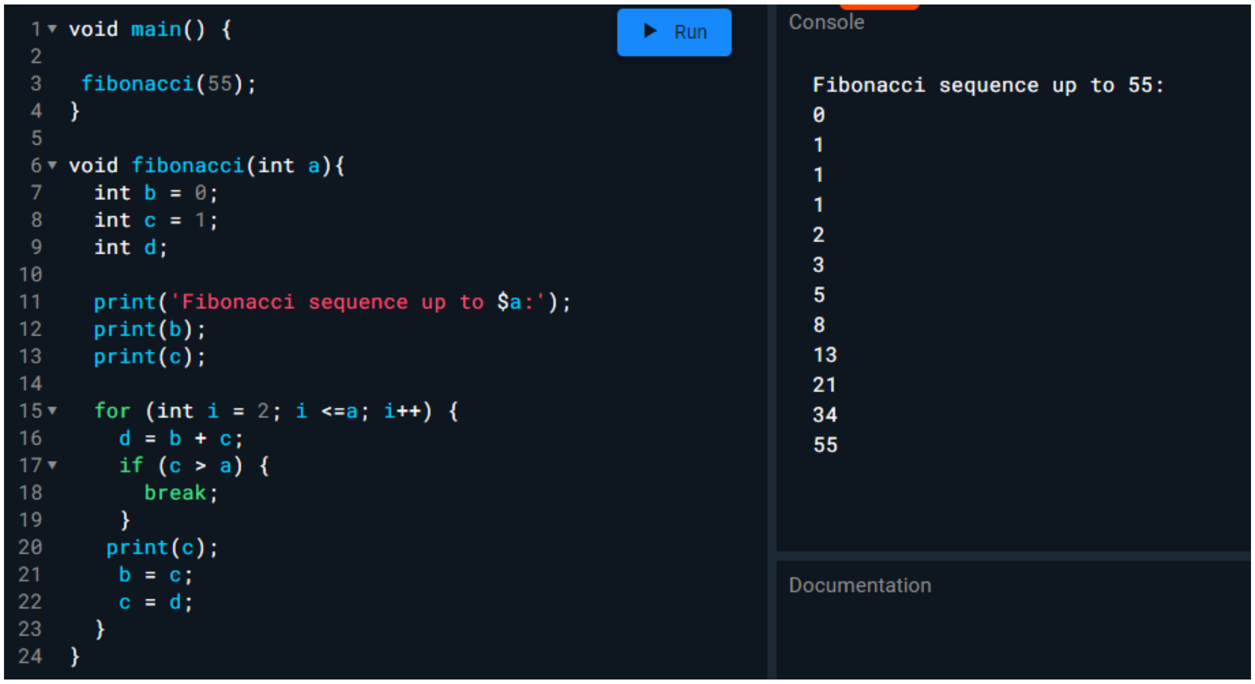


```
void main()
{
  factorial(8);
}
void factorial(int a)
{
  int fact=1;
  for(int i=1; i<=a; i++)
  {
    fact=fact*i;
  }
 print("The factorial of $a is $fact");
}
```

```
The factorial of 8 is 40320
```

3. Write a program to print out the Fibonacci sequence up to a given number.

```
void main() {

 fibonacci(55);
}

void fibonacci(int a){
  int b = 0;
  int c = 1;
  int d;

  print('Fibonacci sequence up to $a:');
  print(b);
  print(c);

  for (int i = 2; i <=a; i++) {
    d = b + c;
    if (c > a) {
      break;
    }
   print(c);
    b = c;
    c = d;
  }
}
```

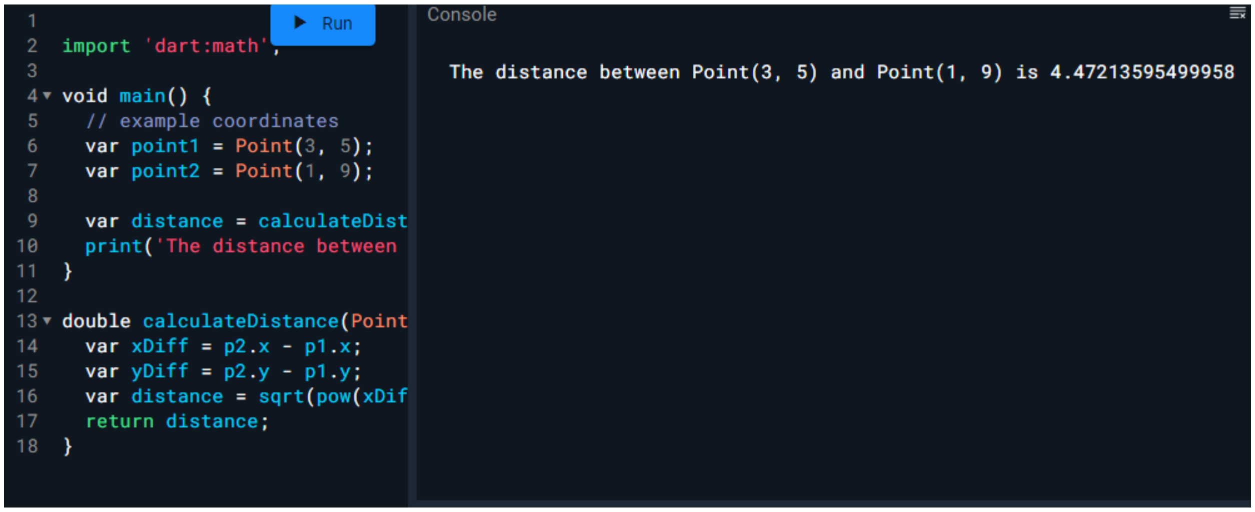


```
void main() {

 fibonacci(55);
}

void fibonacci(int a){
  int b = 0;
  int c = 1;
  int d;

  print('Fibonacci sequence up to $a:');
  print(b);
  print(c);

  for (int i = 2; i <=a; i++) {
    d = b + c;
    if (c > a) {
      break;
    }
   print(c);
    b = c;
    c = d;
  }
}
```

```
Fibonacci sequence up to 55:
0
1
1
1
2
3
5
8
13
21
34
55
```

4. Write a program to calculate the distance between two points on a 2D plane using a function.

```
import 'dart:math';
void main() {
  // example coordinates
  var point1 = Point(3, 5);
  var point2 = Point(1, 9);

  var distance = calculateDistance(point1, point2);
  print('The distance between $point1 and $point2 is $distance');
}
double calculateDistance(Point p1, Point p2) {
  var xDiff = p2.x - p1.x;
  var yDiff = p2.y - p1.y;
  var distance = sqrt(pow(xDiff, 2) + pow(yDiff, 2));
  return distance;
}
```

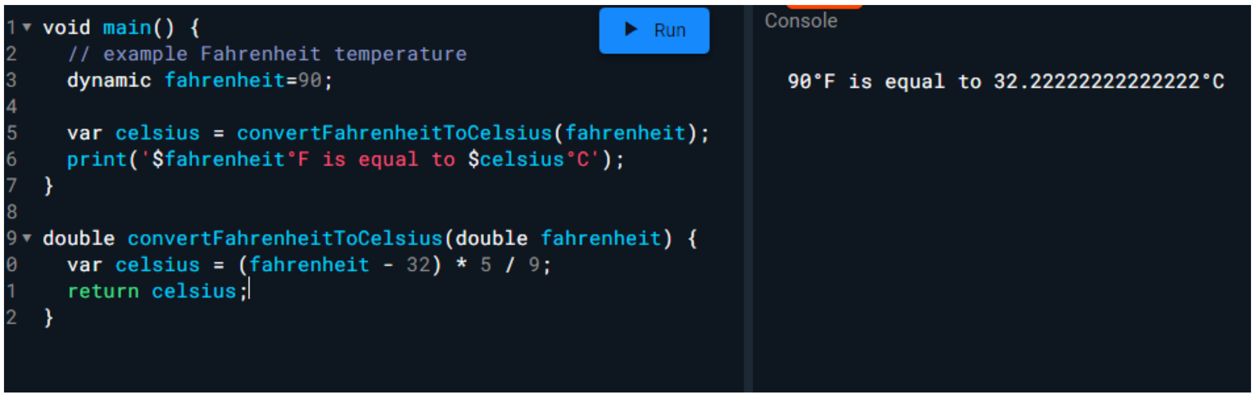


```

import 'dart:math';

void main() {
  // example coordinates
  var point1 = Point(3, 5);
  var point2 = Point(1, 9);

  var distance = calculateDist
  print('The distance between
}

double calculateDistance(Point
  var xDiff = p2.x - p1.x;
  var yDiff = p2.y - p1.y;
  var distance = sqrt(pow(xDif
  return distance;
}
```

```
The distance between Point(3, 5) and Point(1, 9) is 4.47213595499958
```

5. Write a program to convert a temperature from Fahrenheit to Celsius using a function

```
void main() {
  // example Fahrenheit temperature
  dynamic fahrenheit=90;

  var celsius = convertFahrenheitToCelsius(fahrenheit);
  print('$fahrenheit°F is equal to $celsius°C');
}

double convertFahrenheitToCelsius(double fahrenheit) {
  var celsius = (fahrenheit - 32) * 5 / 9;
  return celsius;
}
```

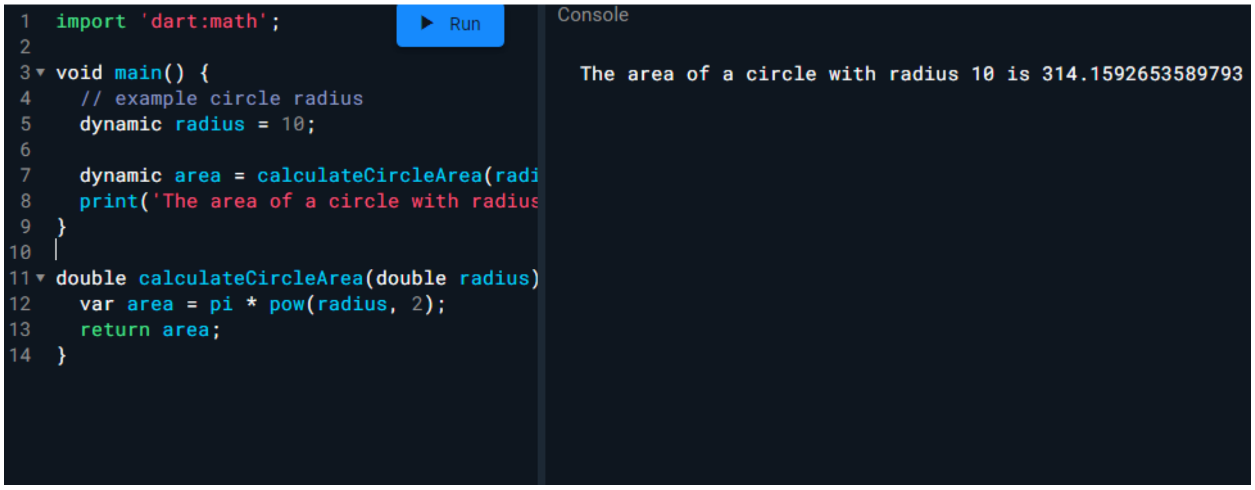


```
void main() {
  // example Fahrenheit temperature
  dynamic fahrenheit=90;

  var celsius = convertFahrenheitToCelsius(fahrenheit);
  print('$fahrenheit°F is equal to $celsius°C');
}

double convertFahrenheitToCelsius(double fahrenheit) {
  var celsius = (fahrenheit - 32) * 5 / 9;
  return celsius;
}
```

```
90°F is equal to 32.22222222222222°C
```

6. Write a program to calculate the area of a circle using a function.

```
import 'dart:math';

void main() {
  // example circle radius
  dynamic radius = 10;

  var area = calculateCircleArea(radius);
  print('The area of a circle with radius $radius is $area');
}

double calculateCircleArea(double radius) {
  var area = pi * pow(radius, 2);
  return area;
}
```



```
import 'dart:math';

void main() {
  // example circle radius
  dynamic radius = 10;

  dynamic area = calculateCircleArea(radi
  print('The area of a circle with radius
}

double calculateCircleArea(double radius)
  var area = pi * pow(radius, 2);
  return area;
}
```

```
The area of a circle with radius 10 is 314.1592653589793
```

7. Write a program to print out the prime numbers between 1 and a given number.

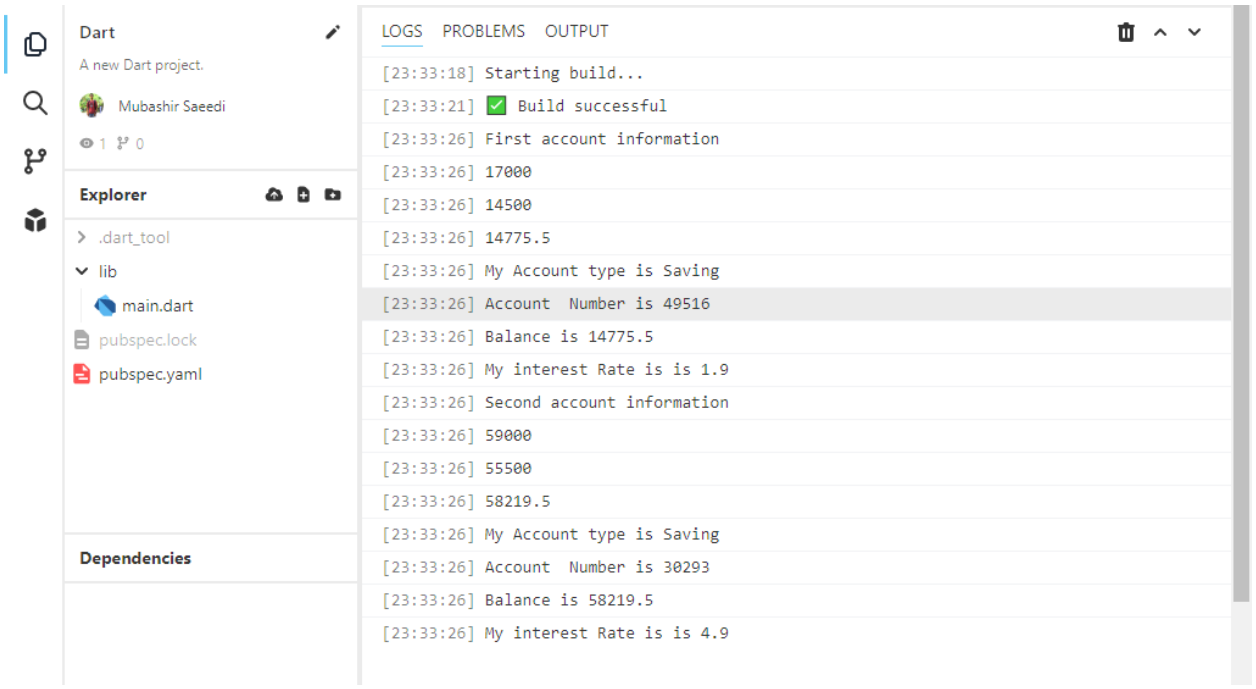
```
import 'dart:math';
void main() {
  // example upper limit
  var limit = 30;

  print('The prime numbers between 1 and $limit are:');
  for (var i = 2; i <= limit; i++) {
    if (isPrime(i)) {
      print(i);
    }
  }
}

bool isPrime(int number) {
  if (number <= 1) {
    return false;
  }

  for (var i = 2; i <= sqrt(number); i++) {
    if (number % i == 0) {
      return false;
    }
  }
  return true;
}
```

Question:1

Create a class called "BankAccount" with the following attributes:
account_number (integer)
balance (double)
account_type (string)
interest_rate (double)
And the following methods:
deposit(amount): adds the amount to the balance.
withdraw(amount): subtracts the amount from the balance. You cannot withdraw more than the current balance.
add_interest(): adds interest to the balance based on the interest rate.
display(): prints out the account number, balance, account type, and interest rate.
Then, create two instances of the BankAccount class, each with its own account number, balance, account type, and interest rate.
Finally, call the deposit(), withdraw(), add_interest(), and display() methods on each instance and confirm that the information is updated and displayed correctly.

```
void main()
{

  BankAccount account1 = BankAccount("Saving",49516,10000,1.9);
  BankAccount account2 = BankAccount("Saving",30293,50000,4.9);

  print("First account information");
  account1.deposit(7000);
  account1.withdraw(2500);
  account1.addInterest();
  account1.display();

  print("Second account information");
  account2.deposit(9000);
  account2.withdraw(3500);
  account2.addInterest();
  account2.display();

}

class BankAccount{

  String accountType;
  int accountNumber;
  double balance;
  double interestRate;

BankAccount(this.accountType,this.accountNumber,this.balance,this.interestRate);

void deposit(double amount)
{
  balance = balance + amount;
  print(balance);
}

void withdraw(double amount)
{
  if(amount <= balance){
  balance = balance - amount;
  print(balance);
  }
  else{
    print("You cannot withdraw more than the current balance.");
  }
}

void addInterest()
{
  double interest = ( balance * (interestRate/100));
  balance = balance + interest;
  print(balance);
}
void display(){
  print("My Account type is $accountType");
  print("Account  Number is $accountNumber");
  print("Balance is $balance");
  print("My interest Rate is is $interestRate");
}
```

}



```
[23:33:18] Starting build...
[23:33:21] ✅ Build successful
[23:33:26] First account information
[23:33:26] 17000
[23:33:26] 14500
[23:33:26] 14775.5
[23:33:26] My Account type is Saving
[23:33:26] Account  Number is 49516
[23:33:26] Balance is 14775.5
[23:33:26] My interest Rate is is 1.9
[23:33:26] Second account information
[23:33:26] 59000
[23:33:26] 55500
[23:33:26] 58219.5
[23:33:26] My Account type is Saving
[23:33:26] Account  Number is 30293
[23:33:26] Balance is 58219.5
[23:33:26] My interest Rate is is 4.9
```

Question:2

Create a class called "Student" with the following attributes:

name (string)

id (string)

courses (list of strings)

And the following methods:

add_course(course): adds a course to the student's list of courses.

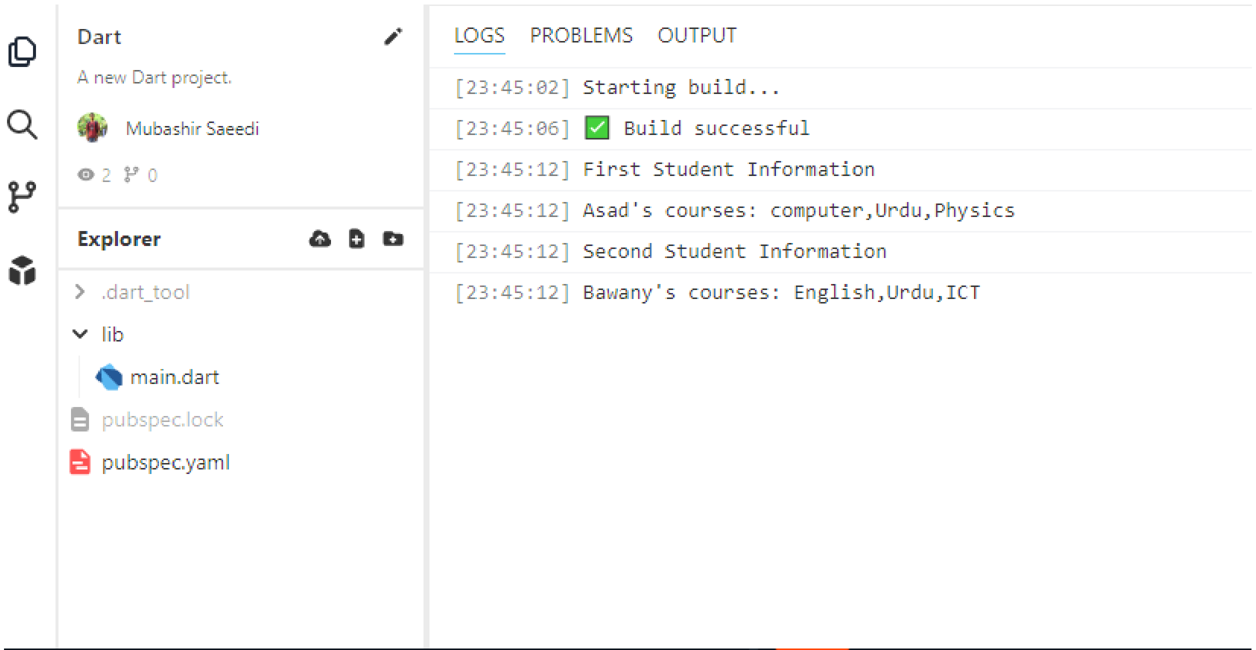
drop_course(course): removes a course from the student's list of courses.

display_courses(): prints out the student's list of courses.

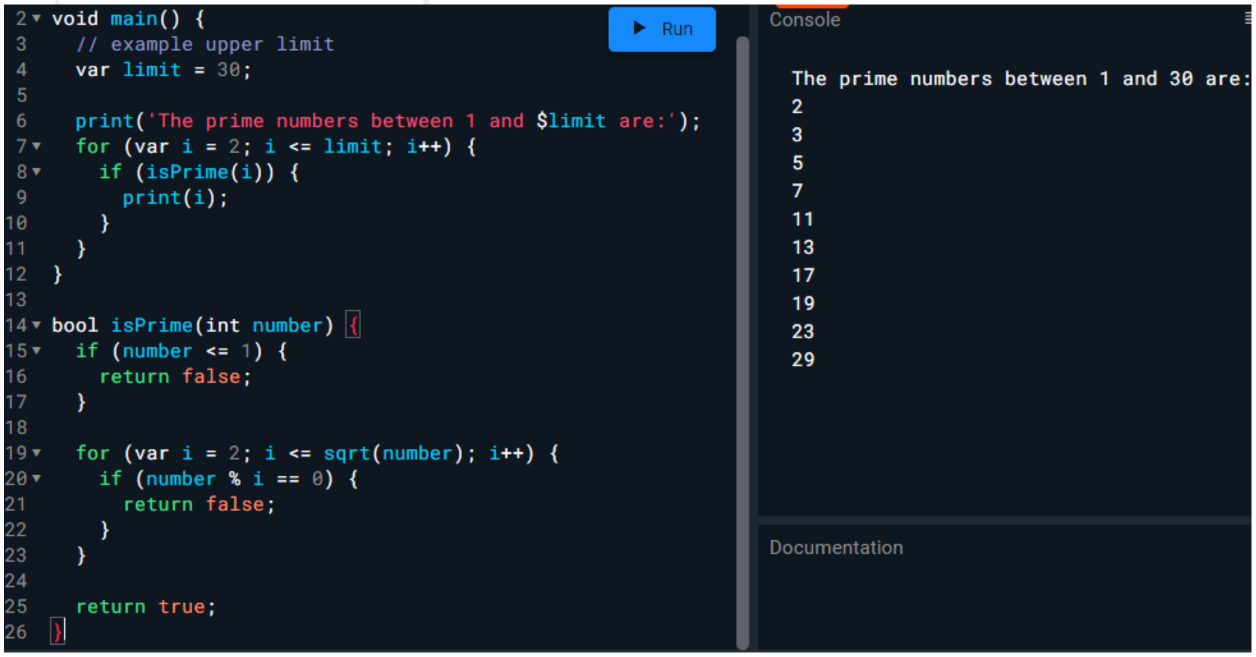
Then, create two instances of the Student class, each with their name, id, and courses.

Finally, call the add_course(), drop_course(), and display_courses() methods on each instance and confirm that the information is updated and displayed correctly.

```
void main(List<String> args)
{
 print("First Student Information");

 Student stud1 = Student("Asad","201A-F22-005",["computer,Urdu"]);
 stud1.add_course("Physics");
 stud1.drop_course("Chemistry");
 stud1.display_courses();

 print("Second Student Information");
 Student stud2 = Student("Bawany","BSE-22S-082",["English,Urdu"]);
 stud2.add_course("ICT");
 stud2.drop_course("DSA");
 stud2.display_courses();
}

class Student
{

 String name;
 String id;
 List <String> courses;

 Student(this.name,this.id,this.courses);

 void add_course(String course)
 {
  courses.add(course);
 }

 void drop_course(String course)
 {
  courses.remove(course);
 }

 void display_courses()
 {
  print('${name}\'s courses: ${courses.join(",")}');
 }

 }
```



```
[23:45:02] Starting build...
[23:45:06] ✅ Build successful
[23:45:12] First Student Information
[23:45:12] Asad's courses: computer,Urdu,Physics
[23:45:12] Second Student Information
[23:45:12] Bawany's courses: English,Urdu,ICT
```



```dart
void main() {
  // example upper limit
  var limit = 30;

  print('The prime numbers between 1 and $limit are:');
  for (var i = 2; i <= limit; i++) {
    if (isPrime(i)) {
      print(i);
    }
  }
}

bool isPrime(int number) {
  if (number <= 1) {
    return false;
  }

  for (var i = 2; i <= sqrt(number); i++) {
    if (number % i == 0) {
      return false;
    }
  }

  return true;
}
```

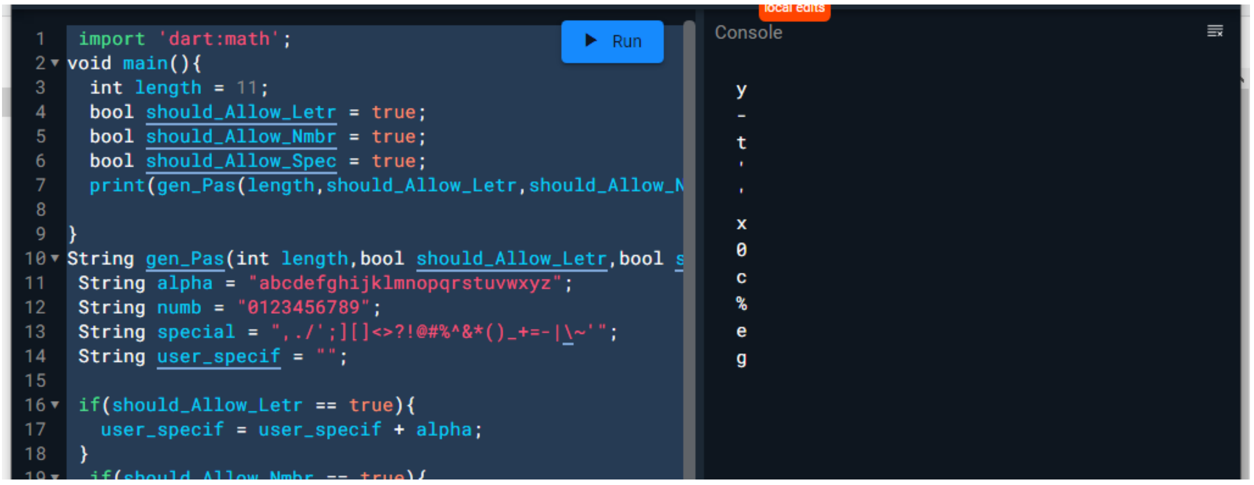
```
The prime numbers between 1 and 30 are:
2
3
5
7
11
13
17
19
23
29
```

Class Quiz # 01

Q#1:

(a)Write a program that generates a random password based on the user's specifications. The user should be able to specify the length of the password and whether it should include numbers, letters, and special characters.

```
import 'dart:math';

void main(){

  int length = 11;

  bool should_Allow_Letr = true;

  bool should_Allow_Nmbr = true;

  bool should_Allow_Spec = true;

  print(gen_Pas(length,should_Allow_Letr,should_Allow_Nmbr,should_Allow_Spec));


}

String gen_Pas(int length,bool should_Allow_Letr,bool should_Allow_Nmbr,bool should_Allow_Spec){

 String alpha = "abcdefghijklmnopqrstuvwxyz";

 String numb = "0123456789";

 String special = ",./';][]<>?!@#%^&*()_+=-|\~'";

 String user_specif = "";


 if(should_Allow_Letr == true){

   user_specif = user_specif + alpha;

 }

  if(should_Allow_Nmbr == true){

   user_specif = user_specif + numb;

 }

   if(should_Allow_Spec == true){

   user_specif = user_specif + special;
 }

  String password = "";
    for(int i = 0; i<length; i++){
      int index =  Random().nextInt(user_specif.length);
      print(user_specif[index]);
    }
    return password;
}
```



```
 import 'dart:math';
void main(){
  int length = 11;
  bool should_Allow_Letr = true;
  bool should_Allow_Nmbr = true;
  bool should_Allow_Spec = true;
  print(gen_Pas(length,should_Allow_Letr,should_Allow_N

}
String gen_Pas(int length,bool should_Allow_Letr,bool s
 String alpha = "abcdefghijklmnopqrstuvwxyz";
 String numb = "0123456789";
 String special = ",./';][]<>?!@#%^&*()_+=-|\~'";
 String user_specif = "";

 if(should_Allow_Letr == true){
   user_specif = user_specif + alpha;
 }
  if(should_Allow_Nmbr == true){
```

```
y
-
t
'
,
x
0
c
%
e
g
```

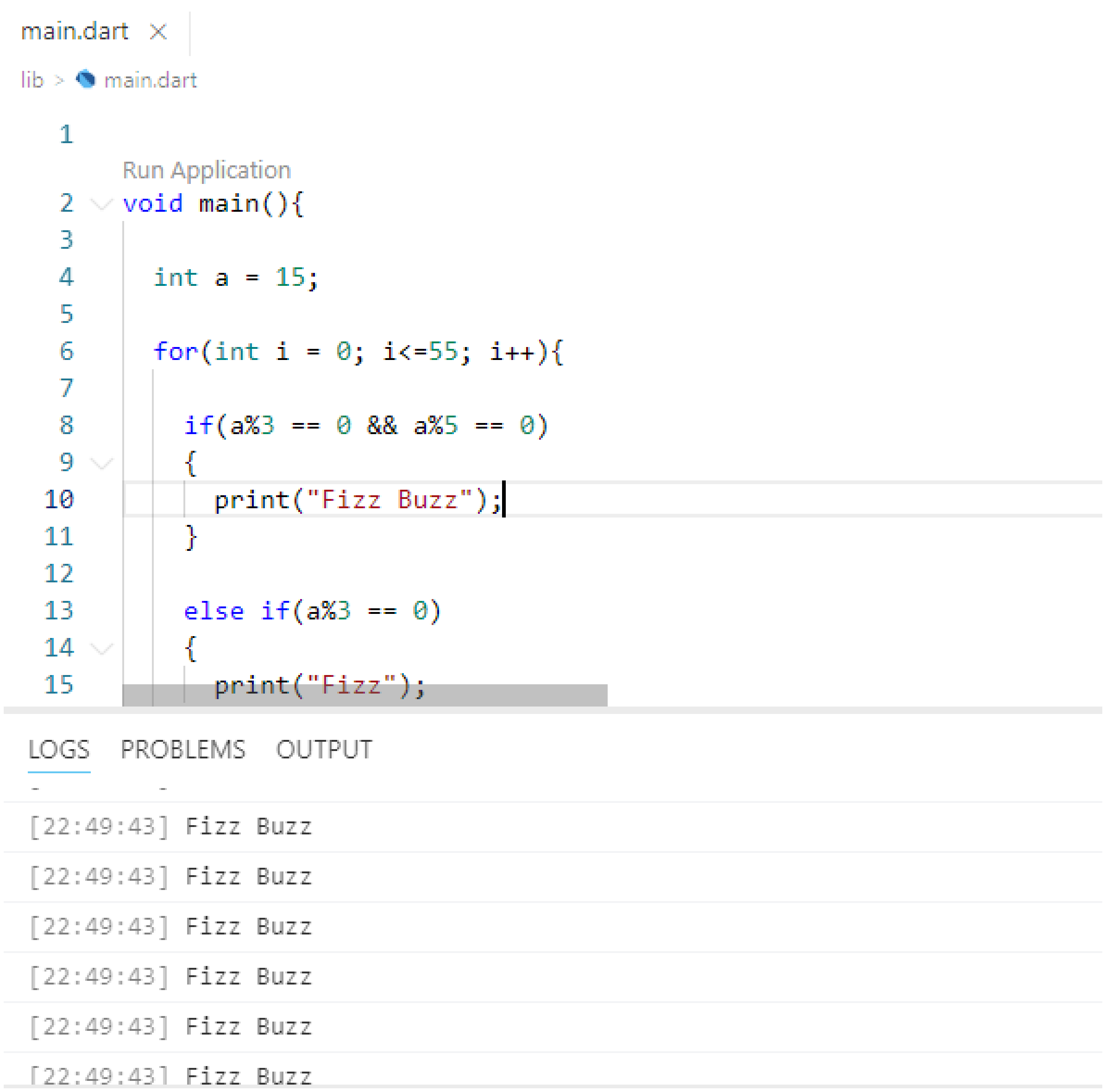
(b)

Write a function that takes a string input from the user and checks if the password is strong enough based on certain criteria (e.g. minimum length, use of uppercase letters, use of special characters, etc.).

Q#2:

Write a program that takes an integer input from the user and prints out numbers from 1 to that integer, but for multiples of 3 print "Fizz" instead of the number, and for multiples of 5 print "Buzz". For numbers that are multiples of both 3 and 5, print "FizzBuzz".

```
void main(){

  int a = 15;

  for(int i = 0; i<=55; i++){
    if(a%3 == 0 && a%5 == 0)
    {
      print("Fizz Buzz");
    }

    else if(a%3 == 0)
    {
      print("Fizz");
    }

    else if(a%5 == 0)
    {
      print("Buzz");
    }

  }

}
```



```dart

void main(){

  int a = 15;

  for(int i = 0; i<=55; i++){

    if(a%3 == 0 && a%5 == 0)
    {
      print("Fizz Buzz");
    }

    else if(a%3 == 0)
    {
      print("Fizz");
```

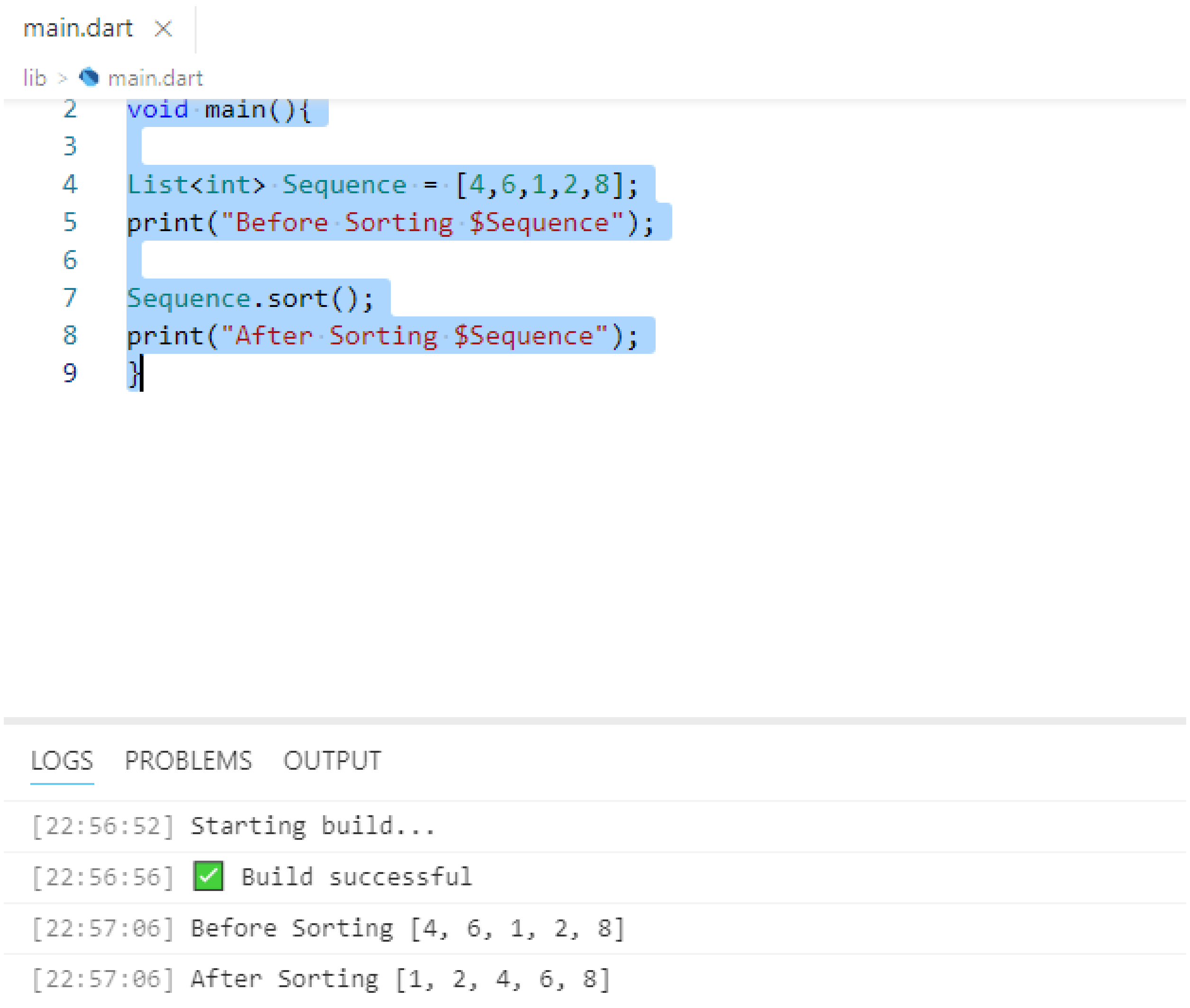
```
[22:49:43] Fizz Buzz
[22:49:43] Fizz Buzz
[22:49:43] Fizz Buzz
[22:49:43] Fizz Buzz
[22:49:43] Fizz Buzz
[22:49:43] Fizz Buzz
```

Q#3:

Write a function that takes a list of numbers as input and sorts the list in ascending order.

```dart
void main(){

List<int> Sequence = [4,6,1,2,8];
print("Before Sorting $Sequence");

Sequence.sort();
print("After Sorting $Sequence");
```

}



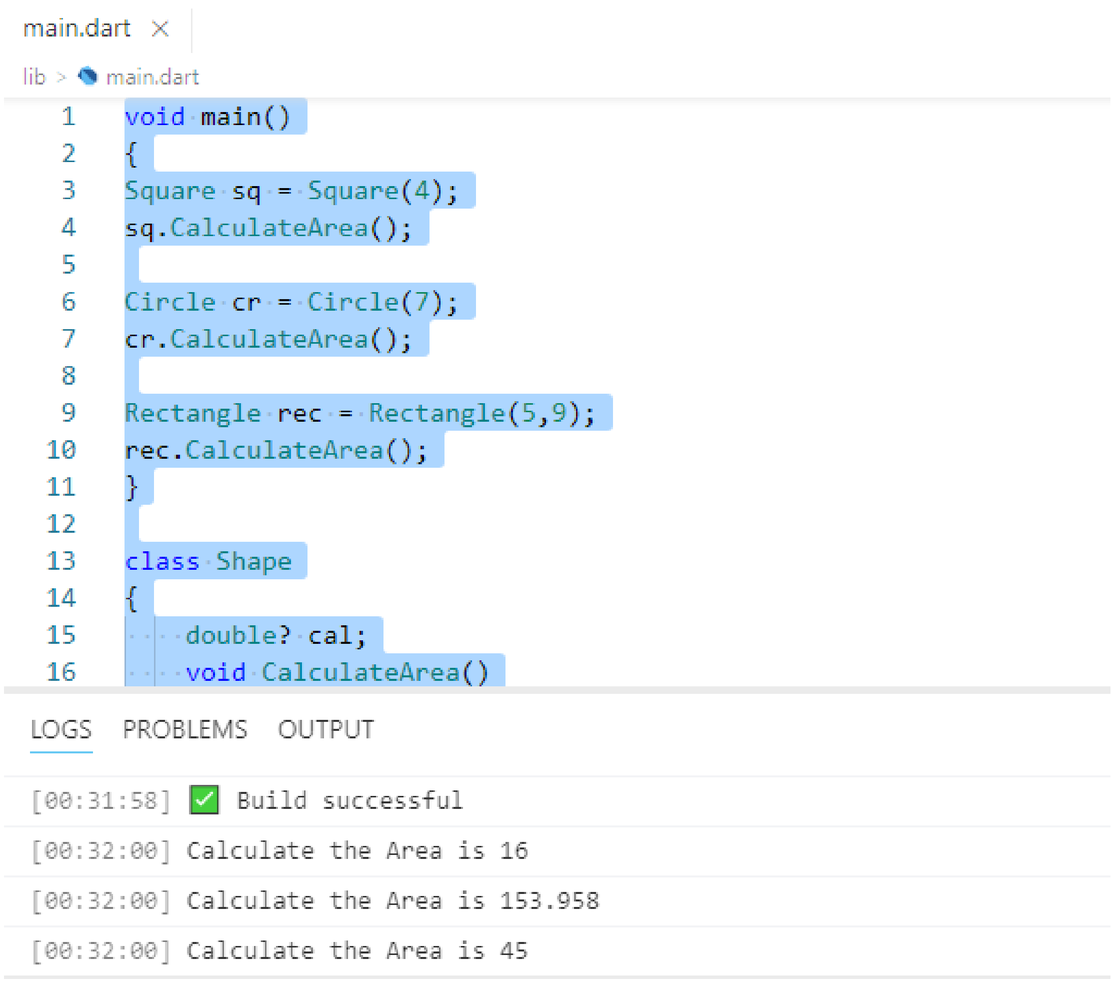
```
void main(){

List<int> Sequence = [4,6,1,2,8];
print("Before Sorting $Sequence");

Sequence.sort();
print("After Sorting $Sequence");
}
```

```
[22:56:52] Starting build...
[22:56:56] ✅ Build successful
[22:57:06] Before Sorting [4, 6, 1, 2, 8]
[22:57:06] After Sorting [1, 2, 4, 6, 8]
```

TASK INHERITANCE

Calculate area of different shapes

```
void main()
{
Square sq = Square(4);
sq.CalculateArea();

Circle cr = Circle(7);
cr.CalculateArea();

Rectangle rec = Rectangle(5,9);
rec.CalculateArea();
}

class Shape
{
    double? cal;
    void CalculateArea()
    {
     print("Calculate the Area is $cal");
    }
}

class Square extends Shape
{
  double sqr;
  Square(this.sqr);
  @override
  void CalculateArea()
  {
  cal = sqr*sqr;
  super.CalculateArea();
  }
}

class Circle extends Shape
{
  int cr;
  Circle(this.cr);
  @override
  void CalculateArea()
  {
  cal = 3.142*cr*cr;
  super.CalculateArea();
  }
}

class Rectangle extends Shape
{
  double len;
  double bre;
  Rectangle(this.len,this.bre);
  @override
  void CalculateArea()
  {
  cal = (len*bre);
  super.CalculateArea();
  }
}
```



main.dart

lib > main.dart

```
void main()
{
Square sq = Square(4);
sq.CalculateArea();

Circle cr = Circle(7);
cr.CalculateArea();

Rectangle rec = Rectangle(5,9);
rec.CalculateArea();
}

class Shape
{
    double? cal;
    void CalculateArea()
```

LOGS PROBLEMS OUTPUT

```
[00:31:58] ✅ Build successful
[00:32:00] Calculate the Area is 16
[00:32:00] Calculate the Area is 153.958
[00:32:00] Calculate the Area is 45
```

Concept of Constructor

--Main Page-----------------------------------

```
import 'student.dart';
void main(){
  Student info = Student();
  info.information();
}
```

--Other Page-----------------------------------

```
class Student{
  String? _name;
  int? roll;

void information(){
  this._name = "Asad";
  this. roll = 59;
  print(_name);
  print(roll);
}
}
```

Invoked a Constructor by default

^_

--Main Page-----------------------------------

```
import 'student.dart';
void main(){
  Student info = Student("Mubashir",16);
  info.information();
}
```

--Other Page----------------------------------=

```
class Student{
  String? _name;
  int? roll;

  Student(this._name, this.roll);

void information(){
  print(_name);
  print(roll);
}

}
```

Create a Parameterized Constructor by user

^_

-----------------------------Main Page---------------------------

```
import 'student.dart';


void main(){
  Student info = Student(name:"Mubashir",roll:84);
  info.information();
}
```

-------------------------------Other Page--------------------------

```
class Student{
  String? name;
  int? roll;

Student({this.name, this.roll});

void information(){
  print(name);
  print(roll);
}
}
```

Create a Name Constructor by user

^_

------------------------------Main Page---------------------------

```
import 'student.dart';

void main(){
  Student info = Student("mubashir",contact:"03432949516",roll:45);

  info.information();
}
```

--------------------------------Other Page--------------------------

```
class Student{
  String? _name;
  int? roll;
  String contact;
Student(this._name,{required this.contact, this.roll=54});

void information(){
  print(_name);
  print(roll);
  print(contact);
}
}
```

Create a Optional Constructor by user

^_