

Introduction to flutter

- Advantages of flutter
- Dart has a large repository of software packages which lets you to extend the capabilities of your application
- Developers need to write just a single code base for both applications (both Android and iOS platforms).
- Flutter needs lesser testing.
- Flutter's simplicity makes it a good candidate for fast development.
- With Flutter, developers has full control over the Widgets and its layout.
- Flutter offers great developer tools, with amazing hot reload.

- Disadvantages of Flutter

- Since it is coded in Dart language, a developer needs to learn new language (though it is easy to learn).
- Modern framework tries to separate logic and UI as much as possible but, in Flutter, user interface and logic is intermixed. We can overcome this using smart coding and using high level module to separate user interface and logic.

- Flutter is yet another framework to create mobile application. Developers are having a hard time in choosing the right development tools in hugely populated segment.

Introduction to Dart programming

- Dart is an Open-Source general-purpose programming language.
- It is originally developed by Google.
- Dart is an Object-Oriented Language with C-Style Syntax.
- It supports programming concepts like interfaces, classes, unlike other programming languages Dart doesn't support arrays directly.
- Dart Collections can be used to replicate data structures such as arrays, generics, and optional typing.

Simple Dart program

```
Void main() {
```

```
    Print("Dart language is easy to learn");  
}
```

- Variables and Data types

Variable is named storage location and Data types simply refers to the type and size of data associated with variables and functions.

Teacher's Signature with Date : _____

Dart uses var keyword to declare the Variable.
The syntax of var is defined below,

Var name = 'Dart';

Constants

The final and const keyword are used to declare Constants.

Void main() {

final a = 12;

const pi = 3.14;

Print(a);

Print(pi);

}

Data types

- Numbers - It is used to represent numeric literals
- Integer and Double
- strings - It represents a sequence of characters. String values are specified in either Single or double quotes.
- Booleans - Dart uses the bool keyword to represent Boolean values - true and false.
- List and Maps - It is used to represent a collection of objects.

Lists

```
void main() {  
    var list = [1, 2, 3, 4, 5];  
    Print(list);  
}
```

Map

```
void main() {  
    var mapping = {'id': 1,  
                  'name': 'Bart'};  
    Print(mapping);  
}
```

Dynamic

```
void main() {  
    dynamic name = "Dart";  
    print(name);  
}
```

Decision Making and Loops

- A decision making block evaluates a condition before the instructions are exuted. Dart supports if, if...else and switch statements.
- Loops are used to repeat a block of code until a specific condition is met. Dart supports for, for ..in, while and do..while loops.

```
void main() {  
    for (var i = 1; i <= 10; i++) {  
        if (i % 2 == 0) {  
            }
```

}

}

}

Teacher's Signature with Date : _____

Functions

A function is a group of statements that together performs a specific task.

Void main () {

 add (3, 4);

}

Void add(int a, int b) {

 Int c;

 c = a + b;

 print (c);

}

Practical 1

1) ~~Simple Program~~

```
void main(){
    print("Hello world");
}
```

2) Simple Program

```
void main(){
    int num1 = 10;
    double num2 = 10.1;
    bool num3 = true;
    String str1 = "Hello All";
    print(num1);
    print(num2);
    print(num3);
    print(str1);
}
```

* Different types of operators in dart

- 1 Arithmetic Operators
- 2 Relational Operators
- 3 Type test Operators
- 4 Bitwise Operators
- 5 Assignment Operators
- 6 Logical Operators

```
void main(){
    print("Hello world");
}
```

Teacher's Signature with Date : _____

```
1 void main()
2 {
3     int a;
4     int b;
5     var result;
6     print("sum of a and b is $c");
7     var c = a + b;
8     print("the difference between a and b is $d");
9     var d = a - b;
10    print("the product of a and b is $f");
11    var f = a * b;
12    print("the quotient of a and b is $g");
13    var g = a / b;
14    print("the remainder of a and b is $h");
15    var h = a % b;
16 }
```

7 Conditional Operators

8 Cascade Notation

void main() {

int a = 2;

int b = 3;

var c = a + b;

print("sum of a and b is \$c");

var d = a - b;

print("The difference between a and b is \$d");

Var e = -d;

Print("The negation of difference between a and b
is \$e");

Var f = a * b;

print("The product of a and b is \$f");

Var g = b / a;

print("The quotient of a and b is \$g");

Var h = b - a;

print("The quotient of a and b is \$h");

```
void main()
{
    var marks=74;
    if (marks >=90)
    {
        print(" excellent ");
    }
    else if(marks>70)
    {
        print(" very good");
    }
    else if(marks>60)
    {
        print(" good ");
    }
    else
    {
        print(" average");
    }
}
```

good

```
var i = b%a;
print("The remainder of a and b is $i");
}
```

* Decision making program

```
void main()
{
    var marks = 74;
    if (marks > 85)
    {
        print ("Excellent");
    }
    else if (marks > 75)
    {
        print ("Very Good");
    }
    elseif (marks > 65)
    {
        print ("Good");
    }
    else
    {
        print ("Average");
    }
}
```

* Find out the factorial number

```
void main()
{
    print ( factorial(5));
}
```

```
1 void main(){
2     cout<<factorial(5);
3 }
4 factorial (number){
5     if(number==0){
6         return 1;
7     }else{
8         return(number * factorial(number - 1));
9     }
10 }
```

① ■ 720

```
factorial (number){  
    if (number <= 0){  
        return 1;  
    } else {  
        return (number * factorial (number - 1));  
    }  
}
```

* Find out a prime number

```
bool isPrime (N){  
    for(var i=2; i<=N / i; ++i){  
        if (N % i == 0){  
            return false;  
        }  
    }  
    return true;  
}
```

```
void main (){  
    print ("Enter N");  
    int N = 12;  
    if (isPrime (N)){  
        print ("$N is a prime number");  
    } else {  
        print ("$N is not a prime number");  
    }  
}
```

```
class student {  
    var studentName;  
    var studentAge;  
    var studentRollNo;  
    showStdInfo(){  
        print("student name is : ${studentName}");  
        print("student age is : ${studentAge}");  
        print("student roll number is : ${studentRollNo}");  
    }  
}  
void main(){  
    var student student();  
    std .studentName="ABC";  
    std .studentAge=24;  
    std .studentRollNo=90001;  
    std .showStdInfo();  
}
```

student name is ABC
student age is 24
student roll number is 90001

* Making Class

```
class Student {  
    var stdName;  
    var stdAge;  
    var stdRollNo;  
    showStdInfo(){  
        print("student Name is : $stdName");  
        print("student Age is : $stdAge");  
        print("student Roll number is : $stdRollNo");  
    }  
}
```

```
void main(){  
    var std = new Student();  
    std .stdName = "ABC";  
    std .stdAge = 24;  
    std .stdRollNo = 90001;  
    std .showStdInfo();  
}
```

Practical 2

Aim Designing the mobile App to implement different Widgets

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: MyApp(),
  ));
}
```

```
Class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  State<MyApp> createState() => MyAppstate();
}
```

```
class MyAppstate extends State<MyApp> {
  TextEditingController controller1 = TextEditingController();
  TextEditingController controller2 = TextEditingController();
  int num1 = 0, num2 = 0, result = 0;
```

```
Void add() {
  perform Operation((a, b) => a + b);
```

3

```
Void sub() {
  perform Operation((a, b) => a - b);
```

Teacher's Signature with Date :

3

```
void mul(){
```

```
    perform Operation ((a, b) = a * b);
```

}

```
void div(){
```

```
    if (num2 != 0){
```

```
        perform Operation (a, b) => a ~ b;
```

```
    } else {
```

```
        scaffold Messenger.of (context). showSnackBar (
```

```
            snackBar (
```

```
                context : Text ('Cannot divide by zero'),
```

```
                duration : Duration (seconds: 2),
```

```
            ),
```

```
        );
```

33

```
void performOperation (int Function (int, int) operation)
```

```
{
```

```
    setState (()
```

```
        num1 = int.tryParse (controller 1.text) ?? 0;
```

```
        num2 = int.tryParse (controller 2.text) ?? 0;
```

```
        result = operation (num1!, num2!);
```

```
    );
```

```
3
```

@Override

```
void dispose (){
```

```
    controller 1.dispose ();
```

```
    controller 2.dispose ();
```

```
    super.dispose ();
```

3

@override

Widget build (BuildContext context) {

return Scaffold (

appbar: AppBar (

title: Text ('simple calculator'),

backgroundColor: Colors.blue.shade 900,

),

body: SingleChildScrollView (

padding: EdgeInsets.all(20),

child: Column (

children: [

SizedBox (height: 15),

Text (

'Result is : \$ result',

style: TextStyle (fontSize: 20, color: Colors.
blue.shade 700),

),

SizedBox (height: 15),

TextField (

controller: controller1,

keyboardType: TextInputType.number,

decoration: InputDecoration (

labelText: "Enter number",

border: OutlineInputBorder (

borderRadius: BorderRadius.circular (20),

),

),

),

Sized Box (height: 15),

Textfield (

controller: controller 2,

keyboardType: TextInputType.number,

decoration: InputDecoration (

labelText: "Enter number"

border: OutlineInputBorder (

borderRadius: BorderRadius.circular(20);

),

),

),

Sized Box (height: 15),

Row (

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

Elevated Button (

onPressed: () {

add();

controller 1. clear();

controller 2. clear();

},

child: Text ('ADD')

),

Elevated Button (

onPressed: () {

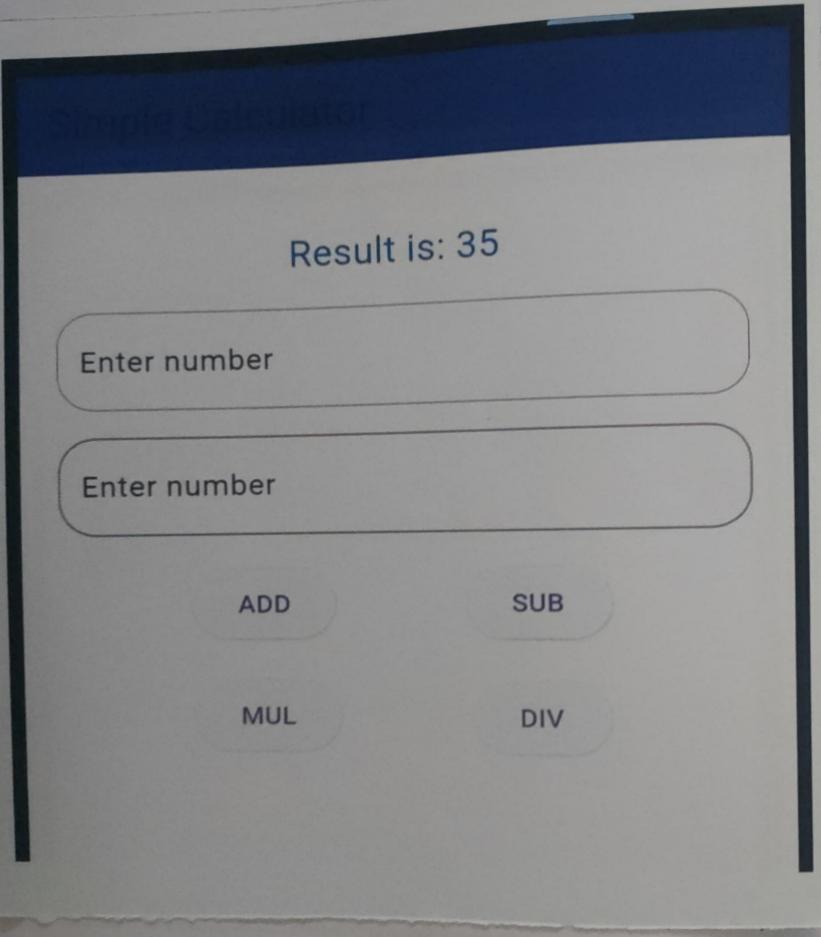
sub();

controller 1. clear();

controller 2. clear();

},

Teacher's Signature with Date : _____



child : Text ('SUB'),
),
],
,
SizedBox (height : 15),
Row (,
mainAxisAlignment : MainAxisAlignment - spaceEvenly ,
children : [
ElevatedButton (,
onPressed : () {
mul ();
Controller 1 . clear ();
Controller 2 . clear ();
},
child : Text ('MUL'),
),
ElevatedButton (,
onPressed : () {
div ();
Controller 1 . clear ();
Controller 2 . clear ();
},
child : Text ('DIV'),
),
],
,
,
,
,
,
,

Practical 3

Aim

Designing the Mobile App to implement different layout

```
import 'package:flutter/material.dart';
void main() {
  runApp(Demoapp());
}

class Demoapp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'My Application',
      debugShowCheckedModeBanner: true,
      home: Scaffold(
        body: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  Container(height: 100, width: 100, color: Colors.teal),
                  Container(
                    height: 100, width: 100, color: Colors.teal[600]),
                  Container(
                    height: 100, width: 100, color: Colors.teal[900]),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Teacher's Signature with Date : _____



1,

2,

Row{

mainAxisAlignment : MainAxisAlignment.spaceEvenly,

children : [

Container{

height : 100, width : 100, color : colors.amberAccent),
width : 100,

color : colors.amberAccent[100]),

Container{height : 100, width : 100,

color : colors.amberAccent[200]),

],

),

],

)

)

);

y

3

Practical 4

Aim

Designing the mobile App to implement the routing

import 'package:flutter/material.dart';

void main() {

 runApp(MaterialApp)

 home: MyApp(),

);

}

class MyApp extends StatelessWidget {

 const MyApp({Key? key}) : super(key: key);

 @override

 Widget build(BuildContext context) {

 TextEditingController name = TextEditingController();

 TextEditingController id = TextEditingController();

 TextEditingController semester = TextEditingController();

 TextEditingController dept = TextEditingController();

 TextEditingController city = TextEditingController();

 return scaffold(

 appBar: AppBar(

 title: Text("user Info"),

 centerTitle: true,

),

 body: Column(

 children: [

 SizedBox(height: 10),

 TextField(

 controller: name, // corrected

 decoration: InputDecoration(

Teacher's Signature with Date : _____

labelText : "Enter your name",
border : OutlineInputBorder()
borderRadius : BorderRadius.circular(15),
,
,
,
SizedBox(height: 10),
TextField(
Controller : id, // corrected
decoration : InputDecoration(
labelText : "Enter your ID",
border : OutlineInputBorder(
borderRadius : BorderRadius.circular(15),
,
,
,
SizedBox(height: 10),
TextField(
Controller : Semester
decoration : InputDecoration(
labelText : "Enter your Semester",
border : OutlineInputBorder(
borderRadius : BorderRadius.circular(15),
,
,
,
SizedBox(height: 10),
TextField(
Controller : dept,

Teacher's Signature with Date : _____

```
decoration : InputDecoration(  
    labelText : "Enter your Department.",  
    border : OutlineInputBorder(  
        borderRadius : BorderRadius.circular(15),  
    ),  
,  
,  
SizedBox(height: 10),  
ElevatedButton(  
    onPressed : () {  
        Navigator.push(  
            context,  
            MaterialPageRoute(  
                builder : (context) => NextScreen(  
                    name : name.text,  
                    id : id.text,  
                    semester : semester.text;  
                    dept : dept.text,  
                    city : city.text,  
                ),  
            ),  
        ).whenComplete(() {  
            name.clear();  
            id.clear();  
            semester = Semester.text  
            dept = dept.text,  
            city = city.text,  
        });  
    },  
);
```

User Info

DEBUG

Enter your name
Kushi Poojary

Enter your ID
2442038

Enter your Semester
3

Enter your Department
IT

Enter your City
Dombivli

Continue

child: Text("Continue"),

)

],

),

);

3

3

Class NeatScreen extends StatelessWidget {
String ? name , id , semester , dept , city ;

NextScreen ({ this.name , this.id , this.semester , this.dept , this.city } ;

@override

Widget build (BuildContext context) {
return scaffold (

body : Column (

children : [

Text (" Name : " + name.toString ()),

Text (" Id : " + id.toString ()),

Text (" Semester : " + semester.toString ()),

Text (" Department : " + dept.toString ()),

Text (" city : " + city.toString ()),

],

),

);

3

Teacher's Signature with Date :

Practical 5

Aim

Designing the mobile app to implement the state management

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: Homescreen(),
  ));
}
```

```
class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);
  @override
  State<HomeScreen> createState() => _HomeScreen
  state();
}
```

```
class HomeScreenState extends State<HomeScreen> {
  TextEditingController name = TextEditingController();
  TextEditingController id = TextEditingController();
  String gendervalue = " ";
  bool hobby1 = false;
  bool hobby2 = false;
  bool hobby3 = false;
  String strhobby1 = " ";
  String strhobby2 = " ";
  String strhobby3 = " ";
  final GlobalKey<FormState> formkey = GlobalKey<FormState>();
```

Teacher's Signature with Date :

@override

Widget build(BuildContext context) {
return scaffold (

appBar : AppBar (

title : Text ("User Info"),

),

body : Form (

key : GlobalKey,

child : Column (

children : [

SizedBox (height: 10),

groupValue : genderValue,

Onchanged : (val) {

setState (());

gender value = val.toString();

});

},

title : Text ("female"),

),

checkboxListTitle (

value : hobby1,

OnChanged : (value) {

setState (());

hobby1 = !hobby1;

if (hobby1) {

strhobby1 = 'playing';

};

});

},

Teacher's Signature with Date : _____

title : Text ("playing"),
,

checkboxlist title (

value : hobby 2,

Onchanged : (value){

setstate (){

hobby2 = ! hobby2;

if (hobby2){

strhobby2 = 'singing';

}

});

},

title : Text ("singing"),

),

checkbox list title (

value : hobby 3,

Onchanged : (value){

setstate (){

hobby3 = ! hobby3;

if (hobby3){

strhobby3 = 'Drawing';

},

});

},

title : Text ("Drawing"),

),

Elevated Button (

Onpressed : (){

if (formKey. currentstate != validate ()) { }

```
if(genderValue != "") {
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => NextScreen(
                name: name.text,
                id: id.text,
                gender: genderValue,
                hobbies: '$strHobby1 $strHobby2 $strHobby3',
            ),
        );
    );
}
}

child: Text("Continue"),
),
),
),
),
),
);
}
}

class NextScreen extends StatelessWidget {
final String? name, id, gender, hobby;
NextScreen({
    this.name,
    this.id,
    this.gender,
```

Teacher's Signature with Date : _____

User Info

Enter Your Name

atharva

Enter your ID

2442030

Male

Female

Playing

Singing

Drawing

Continue

this.hobbies,
});

@override

widget build(BuildContext context) {

return scaffold(

body: column(

children: [

Text("Name: " + name.toString()),

Text("Id: " + id.toString()),

Text("Gender: " + gender.toString()),

Text("Hobbies: " + hobbies.toString()),

],

),

);

3

Teacher's Signature with Date : _____

Practical 6

Aim Designing the mobile app to implement the theming and styling

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: MyApp(),
  ));
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Theming and styling'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            Image.network(
              'https://uploads-ssl.webflow.com/5f841209f4e0034471/6078b650748b858d46ff57f-flutter%20development.png',
              height: 250,
              width: 250,
            ),
          ],
        ),
      ),
    );
}
```

Teacher's Signature with Date : _____

Theming and styling

DEBUG



),
,
,
,
,
,

Teacher's Signature with Date : _____

Practical 7

Aim : Designing the mobile app to implement gesture

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(home: MyApp()));
}

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  MyAppState createState() => MyAppState();
}

class MyAppState extends State<MyApp> {
  int number of Times Tapped = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            Text(
              'Tapped' + number of times tapped.toString() + 'times',
              style: TextStyle(fontSize: 30),
            ),
            GestureDetector(
              onTap: () {
                number of times Tapped++;
              }
            )
          ],
        )
      )
    );
}
```

Teacher's Signature with Date :

Tapped 27 times

TAP HERE

3);
3;
child : Container {
padding : Edge Insets.all(20)
color : Colors.green[200],
child : Text {
'TAP HERE',
style : TextStyle(fontsize:30),
)
)
)
)
)
)
)
)
2
3