



# Owl-M-A-Material-Design- Study-App

- ▶ **Owl-M: A Material Design Study App** is a sample application developed by Google to demonstrate the use of **Material Design** principles and best practices in Android development. The app serves as a practical example for developers aiming to learn and implement Material Design components and themes in their Android applications.

- ▶ **Key Features:**

- 1. **Material Design Integration:**

1. **Theming:** Demonstrates the use of Material Theming, including custom colors, typography, and shapes.
2. **Animations:** Utilizes Material Motion for smooth transitions and interactions, such as shared element transitions, fade animations, and transformations.
3. **Components:** Implements various Material Design components like **Bottom Navigation**, **Floating Action Buttons (FABs)**, **AppBar**, **Cards**, and **Snackbars**.

- 2. **Modern Android Development Practices:**

1. **Jetpack Libraries:** Utilizes Android Jetpack libraries such as **Navigation**, **ViewModel**, **LiveData**, and **Data Binding**.
2. **MVVM Architecture:** Follows the Model-View-ViewModel (MVVM) architecture for clean separation of concerns and better testability.
3. **Kotlin Coroutines:** Demonstrates asynchronous data handling using Kotlin coroutines.

## 1.Educational Focus:

- The app's content is centered around educational material, such as courses and lessons,
- providing a realistic context for the demonstration of UI patterns.
- Features a modular course catalog with individual lesson details.

## 2.Responsive Design:

- Implements responsive layouts and adaptive UI components,
- making the app suitable for different screen sizes, including tablets.
- Uses **ConstraintLayout** and **Compose** to showcase modern layout techniques.


### Use Case:

- Developers and Learners:** The app is designed as a learning tool for developers interested in mastering Material Design in Android.
- It serves as a comprehensive example of how to implement Material Design patterns and practices in a functional,
- educational app.

### Tech Stack:

- Languages:** Kotlin
- Libraries:** Android Jetpack, Material Design Components
- Architecture:** MVVM (Model-View-ViewModel)
- UI:** Compose, XML layouts with Material Components

**Owl-M** is part of Google's efforts to provide developers with clear, well-documented examples of how to implement Material Design in modern Android applications.

- 
- ▶ `import android.content.Context`
  - ▶ `import android.content.Intent`
  - ▶ `import android.os.Bundle`
  - ▶ `import androidx.activity.ComponentActivity`
  - ▶ `import androidx.activity.compose.setContent`
  - ▶ `import androidx.compose.foundation.Image`
  - ▶ `import androidx.compose.foundation.background`
  - ▶ `import androidx.compose.foundation.layout.*`
  - ▶ `import androidx.compose.material.*`
  - ▶ `import androidx.compose.runtime.*`
  - ▶ `import androidx.compose.ui.Alignment`
  - ▶ `import androidx.compose.ui.Modifier`
  - ▶ `import androidx.compose.ui.graphics.Color`
  - ▶ `import androidx.compose.ui.layout.ContentScale`
  - ▶ `import androidx.compose.ui.res.painterResource`
  - ▶ `import androidx.compose.ui.text.font.FontFamily`
  - ▶ `import androidx.compose.ui.text.font.FontWeight`
  - ▶ `import androidx.compose.ui.text.input.PasswordVisualTransformation`
  - ▶ `import androidx.compose.ui.tooling.preview.Preview`
  - ▶ `import androidx.compose.ui.unit.dp`
  - ▶ `import androidx.compose.ui.unit.sp`
  - ▶ `import androidx.core.content.ContextCompat`
  - ▶ `import com.example.owlapplication.ui.theme.OwlApplicationTheme`




```
class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            LoginScreen(this, databaseHelper)
        }
    }
}
```

@Composable

```
fun LoginScreen(context: Context, databaseHelper:
UserDatabaseHelper) {
```

```
    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }
```


```
    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
```




```
Image(painterResource(id = R.drawable.study_login),  
contentDescription = "")
```

```
Text(  
    fontSize = 36.sp,  
    fontWeight = FontWeight.ExtraBold,  
    fontFamily = FontFamily.Cursive,  
    text = "Login"  
)  
Spacer(modifier = Modifier.height(10.dp))
```

```
TextField(  
    value = username,  
    onValueChange = { username = it },  
    label = { Text("Username") },  
    modifier = Modifier.padding(10.dp)  
        .width(280.dp)  
)
```



```
TextField(  
    value = password,  
    onValueChange = { password = it },  
    label = { Text("Password") },  
    visualTransformation =  
PasswordVisualTransformation(),  
    modifier = Modifier.padding(10.dp)  
        .width(280.dp)  
)  
  
if (error.isNotEmpty()) {  
    Text(  
        text = error,  
        color = MaterialTheme.colors.error,  
        modifier = Modifier.padding(vertical =  
16.dp)  
    )  
}
```



```
Button(  
    onClick = {  
        if (username.isNotEmpty() &&  
password.isNotEmpty()) {  
            val user =  
databaseHelper.getUserByUsername(username)  
            if (user != null && user.password ==  
password) {  
                error = "Successfully log in"  
                context.startActivity(  
                    Intent(  
                        context,  
                        MainActivity::class.java  
                    )  
                )  
            }  
        }  
    })  
    //onLoginSuccess()  
}  
else {  
    error = "Invalid username or password"  
}  
}
```



```
} else {
    error = "Please fill all fields"
}
},
modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Login")
}
Row {
    TextButton(onClick = {context.startActivity(
        Intent(
            context,
            RegisterActivity::class.java
        )
    )})
    { Text(text = "Register") }
    TextButton(onClick = {
    })


    {
        Spacer(modifier = Modifier.width(60.dp))
        Text(text = "Forget password?")
    }
}
```

```
}  
    }  
}  
}  
private fun startMainPage(context: Context) {  
    val intent = Intent(context, MainActivity::class.java)  
    ContextCompat.startActivity(context, intent, null)  
}
```



```
draw.scalepackage com.example.owlapplication
```

```
import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
draw.scalepackage com.example.owlapplication
import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
```



```
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
```

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            StudyApp(this)
        }
    }
}
```


```
}  
}  
}
```

```
@Composable  
fun StudyApp(context: Context) {  
  
    Column(  
        modifier = Modifier  
            .padding(20.dp)  
            .verticalScroll(rememberScrollState())  
  
    ) {  
  
        Text(text = "Study Material",  
            fontSize = 36.sp,  
            fontWeight = FontWeight.Bold,  
            color = Color(0xFFFFFA500),  
            modifier =  
Modifier.align(Alignment.CenterHorizontally))
```



Spacer(modifier = Modifier.height(20.dp))

```
//      01
Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
            context.startActivity(
                Intent(context,
MainActivity2::class.java)
            )
        },
    elevation = 8.dp
)
{
    Column(
        horizontalAlignment =
Alignment.CenterHorizontally
    ) {
```




```
Image(
    painterResource(id = R.drawable.img_1),
    contentDescription = "",
    modifier = Modifier
        .height(150.dp)
        .scale(scaleX = 1.2F, scaleY = 1F)
)
Text(text = stringResource(id =
R.string.course1), color = Color(0xFFFFA500),
    fontSize = 16.sp)

Text(
    text = stringResource(id = R.string.topic1),
    fontWeight = FontWeight.Bold,
    fontSize = 20.sp,
    textAlign = TextAlign.Center,
)
}
}

Spacer(modifier = Modifier.height(20.dp))
```

02

```
Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
            context.startActivity(
                Intent(context,
MainActivity3::class.java)
            )
        },
    elevation = 8.dp
) {
    Column(
        horizontalAlignment =
Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.img_2),
            contentDescription = "",
            modifier = Modifier
```



```
.height(150.dp)
        .scale(scaleX = 1.4F, scaleY = 1F)
    )
    Text(text = stringResource(id =
R.string.course2),color = Color(0xFFFFFA500),
        fontSize = 16.sp)

    Text(
        text = stringResource(id = R.string.topic2),
        fontWeight = FontWeight.Bold,
        fontSize = 20.sp,
        textAlign = TextAlign.Center,
    )
}
}

Spacer(modifier = Modifier.height(20.dp))

//    03
Card(
    modifier = Modifier
```

```
.fillMaxWidth()
    .height(250.dp)
    .clickable {
        context.startActivity(
            Intent(context,
MainActivity4::class.java)

        )
    },
    elevation = 8.dp
)
{
    Column(
        horizontalAlignment =
Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.img_3),
contentDescription = "",
            modifier = Modifier
                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
    }
}
```



```
Text(text = stringResource(id = R.string.course3),color  
= Color(0xFFFFFA500),  
      fontSize = 16.sp)
```


```
Text(  
    text = stringResource(id = R.string.topic3),  
    fontWeight = FontWeight.Bold,  
    fontSize = 20.sp,  
    textAlign = TextAlign.Center,  
    )  
}  
}
```

```
Spacer(modifier = Modifier.height(20.dp))
```

```
// 04  
Card(  
    modifier = Modifier  
        .fillMaxWidth()
```

```
.height(250.dp)
    .clickable {
        context.startActivity(
            Intent(context,
MainActivity5::class.java)

        )
    },
    elevation = 8.dp
)
{
    Column(
        horizontalAlignment =
Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.img_4),
contentDescription = "",
            modifier = Modifier
                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(text = stringResource(id =
R.string.course4),
```



```
color = Color(0xFFFFA500),  
    fontSize = 16.sp)
```

```
Text(  
    text = stringResource(id = R.string.topic4),  
    fontWeight = FontWeight.Bold,  
    fontSize = 20.sp,  
    textAlign = TextAlign.Center,  
)
```

```
}
```

```
}
```




```
package com.example.owlapplication
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
```

```
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import
com.example.owlapplication.ui.theme.OwlApplicati
onTheme
```

```
class MainActivity2 : ComponentActivity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting()
        }
    }
}
@Composable
fun Greeting() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end =
26.dp, bottom = 26.dp)
        .verticalScroll(rememberScrollState())
        .background(Color.White),
        verticalArrangement = Arrangement.Top
    ) {
```






```
Image(  
    painterResource(id = R.drawable.img_1),  
    contentDescription = "",  
    modifier =  
Modifier.align(Alignment.CenterHorizontally)  
    .scale(scaleX = 1.5F, scaleY = 1.5F)  
)
```

```
Spacer(modifier = Modifier.height(60.dp))
```

```
Text(  
    text = stringResource(id = R.string.course1),  
    color = Color(0xFFFFFA500),  
    fontSize = 16.sp,  
    modifier =  
Modifier.align(Alignment.CenterHorizontally)  
)
```

```
Spacer(modifier = Modifier.height(20.dp))
```

```
Text(  
    text = stringResource(id = R.string.course2),  
    color = Color(0xFFFFFA500),  
    fontSize = 16.sp,  
    modifier =  
Modifier.align(Alignment.CenterHorizontally)  
)
```



```
text = stringResource(id = R.string.topic1),
    fontWeight = FontWeight.Bold,
    fontSize = 26.sp,
    modifier =
Modifier.align(Alignment.CenterHorizontally)

)
Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id =
R.string.subheading1_1),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)

Spacer(modifier = Modifier.height(20.dp))

Text(
    text = stringResource(id = R.string.text1_1),
    modifier = Modifier.align(Alignment.Start),
    textAlign = TextAlign.Justify,
    fontSize = 16.sp
```

)

```
    Spacer(modifier = Modifier.height(20.dp))
    Text(
        text = stringResource(id =
R.string.subheading1_2),
        modifier = Modifier.align(Alignment.Start),
        fontSize = 20.sp
    )
```

```
    Spacer(modifier = Modifier.height(20.dp))
```

```
    Text(
        text = stringResource(id = R.string.text1_2),
        modifier = Modifier.align(Alignment.Start),
        textAlign = TextAlign.Justify,
        fontSize = 16.sp
    )
```

```
}
```

```
}
```



```
package com.example.owlapplication
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
```



```
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
```

```
class MainActivity3 : ComponentActivity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting1 ()
        }
    }
}
@Composable
fun Greeting1 () {
    Column(
        modifier = Modifier.padding(start = 26.dp, end =
26.dp, bottom = 26.dp)
        .verticalScroll(rememberScrollState())
        .background(Color.White),
        verticalArrangement = Arrangement.Top
    ) {
```




```
Image(
    painterResource(id = R.drawable.img_2),
    contentDescription = "",
    modifier =
Modifier.align(Alignment.CenterHorizontally)
    .scale(scaleX = 1.2F, scaleY = 1F)
)

Spacer(modifier = Modifier.height(20.dp))

Text(
    text = stringResource(id = R.string.course2),
    color = Color(0xFFFFA500),
    fontSize = 16.sp,
    modifier =
Modifier.align(Alignment.CenterHorizontally)
)

Spacer(modifier = Modifier.height(20.dp))

Text(
    text = stringResource(id = R.string.topic2),
```



```
fontWeight = FontWeight.Bold,  
    fontSize = 26.sp,  
    modifier =  
Modifier.align(Alignment.CenterHorizontally)  
  
    )  
    Spacer(modifier = Modifier.height(20.dp))  
    Text(  
        text = stringResource(id =  
R.string.subheading2_1),  
        modifier = Modifier.align(Alignment.Start),  
        fontSize = 20.sp  
    )  
  
    Spacer(modifier = Modifier.height(20.dp))  
  
    Text(  
        text = stringResource(id = R.string.text2_1),  
        modifier = Modifier.align(Alignment.Start),  
        textAlign = TextAlign.Justify,  
        fontSize = 16.sp  
    )
```

```
Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id =
R.string.subheading2_2),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)
```

```
Spacer(modifier = Modifier.height(20.dp))
```

```
Text(
    text = stringResource(id = R.string.text2_2),
    modifier = Modifier.align(Alignment.Start),
    textAlign = TextAlign.Justify,
    fontSize = 16.sp
)
```

```
}
}
```



```
package com.example.owlapplication
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
```

```
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import
com.example.owlapplication.ui.theme.OwlApplicati
onTheme
```

```
class MainActivity4 : ComponentActivity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting2()
        }
    }
}
@Composable
fun Greeting2() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end =
26.dp, bottom = 26.dp)
        .verticalScroll(rememberScrollState())
        .background(Color.White),
```



```
verticalArrangement = Arrangement.Top  
) {
```

```
    Image(  
        painterResource(id = R.drawable.img_3),  
        contentDescription = "",  
        modifier =  
Modifier.align(Alignment.CenterHorizontally)  
        .scale(scaleX = 1.5F, scaleY = 2F)  
    )
```

```
    Spacer(modifier = Modifier.height(60.dp))
```

```
    Text(  
        text = stringResource(id = R.string.course3),  
        color = Color(0xFFFFFA500),  
        fontSize = 16.sp,  
        modifier =  
Modifier.align(Alignment.CenterHorizontally)  
    )
```

```
    Spacer(modifier = Modifier.height(20.dp))
```

```
Text(
    text = stringResource(id = R.string.topic3),
    fontWeight = FontWeight.Bold,
    fontSize = 26.sp,
    modifier =
Modifier.align(Alignment.CenterHorizontally)

)
Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id =
R.string.subheading3_1),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)

Spacer(modifier = Modifier.height(20.dp))

Text(
    text = stringResource(id = R.string.text3_1),
    modifier = Modifier.align(Alignment.Start),
    textAlign = TextAlign.Justify,
```



```
fontSize = 16.sp
    )

    Spacer(modifier = Modifier.height(20.dp))
    Text(
        text = stringResource(id =
R.string.subheading3_2),
        modifier = Modifier.align(Alignment.Start),
        fontSize = 20.sp
    )

    Spacer(modifier = Modifier.height(20.dp))

    Text(
        text = stringResource(id = R.string.text3_2),
        modifier = Modifier.align(Alignment.Start),
        textAlign = TextAlign.Justify,
        fontSize = 16.sp
    )

}
}
```



```
package com.example.owlapplication
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
```



```
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import
com.example.owlapplication.ui.theme.OwlApplicati
onTheme
```

```
class MainActivity5 : ComponentActivity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting3()
        }
    }
}
@Composable
fun Greeting3() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end =
26.dp, bottom = 26.dp)
        .verticalScroll(rememberScrollState())
        .background(Color.White),
```

```
verticalArrangement = Arrangement.Top
) {

    Image(
        painterResource(id = R.drawable.img_4),
        contentDescription = "",
        modifier =
Modifier.align(Alignment.CenterHorizontally)
        .scale(scaleX = 1.5F, scaleY = 1.5F)
    )

    Spacer(modifier = Modifier.height(60.dp))

    Text(
        text = stringResource(id = R.string.course4),
        color = Color(0xFFFFA500),
        fontSize = 16.sp,
        modifier =
Modifier.align(Alignment.CenterHorizontally)
    )
}
```



## Login

Login

[Register](#)

[Forget password?](#)



## Register

Username

Email

Password

Register

Have an account? [Log in](#)

## Study Material



Arts & Craft  
The Basics of Woodturning



Painting  
An introduction to oil painting



Architecture





## Arts & Craft

# The Basics of Woodturning

## What Is WoodTurning

Woodturning is a form of woodworking involving a lathe. With other kinds of woodworking, the wood is stationary and the tool moves to create cuts.

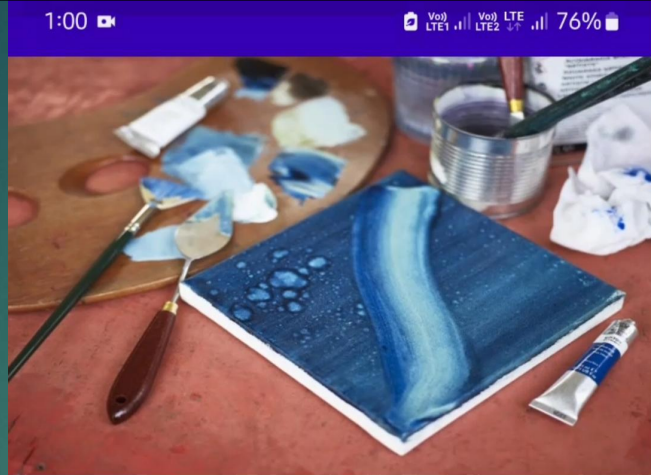
- In woodturning, the lathe turns the wood on its axis at high revolutions per minute while relatively stationary special cutting tools on a tool rest do the work.

A wood lathe allows woodturners to create all kinds of objects, from bowls to stair railings to chess pieces to musical instruments.

## History of Woodturning

The art on monuments in ancient Egypt offers the first recorded instances of spindle turning. These illustrations showed a strap a helper used to rotate the lathe while another worker cut the wood.

The ancient Romans, Chinese, Persians and Arabs had their own variations of the lathe. Early lathe workers would sometimes hold cutting tools with their bare feet while powering the lathe with their hands.



## Painting

### An introduction to oil painting

#### What is oil paint?

- There are three main categories of oil paints: traditional oils, alkyd oils and water-mixable oils. These are all composed of pigment and binder. The binder encapsulates and protects the pigment, while it also acts as an adhesive by attaching neighbouring particles to each other.

#### What ranges do Winsor & Newton have available?

We currently have 4 ranges of oil paint to suit a variety of different practices.

Winsor & Artists Newton' Oil Colour range is a traditional oil paint, it provides the widest choice of colours with the highest pigment strength, ensuring the cleanest, brightest colours and best mixes.

The Winton Oil Colour range is also a traditional oil, but it is formulated to offer dependable quality at an accessible price; it is ideal for first time artists who want to learn the fundamentals of oil painting and



## Architecture

# City Phenomenon between Urban Structure and Composition

## Abstract

Cities are not just a sum of buildings, but especially a set of social relations that their inhabitants develop. Cities are characterized by a wide variety of social groups and lifestyles. An urban composition represents a form of the city in which it gets a formal order, so that the shape of any urban ensemble is not linked to a random phenomenon, but to an intervention mastered and understood as such. For the city, the urban composition represents what the architectural composition represents for a building. This concept regarding the composition is common both to the architecture and to the city. The main property of the composition is that it transforms a possibly dispersed ensemble into a whole, resolving the contradictions that arise when the requirements and conditions of the project are numerous. Spatial forms and urban compositions are built over time, longer than that of architectural composition. On the other hand, "design of the urban environment" is understood by us as a complex formation of public spaces of the city, located on the ground floor level of the city building and ensuring the vital activity of the urban community. This chapter will study the city phenomenon on a large scale.

## Introduction



## Design

# Learning The Basics of Brand Identity

## What Is a Brand Identity?

Is it your logo? Your color palette? Your infographic style? It's all that—and more.

Branding pro Marty Neumeier defines a brand identity as “the outward expression of a brand, including its trademark, name, communications, and visual appearance.” To us, a brand identity is the sum total of how your brand looks, feels, and speaks to people. (Sometimes that even includes how it sounds, tastes, feels, and even smells.)

That said, when most people talk about brand identity, they're referring to a brand's visual identity. For the purposes of this post, that's what we'll be focusing on.

## Why Do You Need a Brand Identity?

A strong brand identity is not about making pretty packaging; it's about communicating your brand story effectively. Design is a powerful tool that can transform how people interact with your brand in three important ways.

1 Differentiation: How can you stand out in a



12:59

VoLTE VoLTE VoLTE 76%



OwlApplication

00:00

