CLASS WORK
SESSIONAL WORK

ASSIGNMENT
EXPERIMENT No 1

SUBMITTED ON .... MARKS OR GRADE OBTAINED ....

NAME Anushka Tomar .... ROLL NO 19100BTCSEMA05474

CLASS .... DEPARTMENT ....

SUBJECT V.R and aug Reality .... CODE NO ....

Signature of Student

Signature of Professor

Experiment - 1

Aim :- AR template

program :-

```
import UIKit
import SceneKit
import ARKit
class View Controller : UIView Controller,
  ARSCNView Delegate {
  @IBOutlet var sceneView : ARSCNView!
  override func view DidLoad () {
   super.viewDidload ()
   sceneView.delegate = self
   Scene View. showsStatistics = true
   let scene = SCNScene (named: "art.scnassets
                                  /ship.scn")!
 } sceneView.scene = scene
  override func viewWillAppear(_animated:Bool) {
   super.viewWill Appear (animated)
   let configuration = AR World Tracking Configu
                                        ration()
   scene View.session.run (configuration)
 }
override func viewWillDisappear (_ animated: Bool){
    super. viewWillDisappear (animated)
    sceneView . session. pause ()
}
func renderer (_ renderer : SCNSceneRenderer,
  node for anchor : AR Anchor) -> SCNNode? {
      let node = SCNNode ()
  } return node
  }

func session (_session : ARSession, didFailWith
                                  Error error : Error){
      } // Present an message to the user
func session wasInterrupted (_session : ARSession)
    {
    }

func sessionInterruptionEnded (_session : ARSession)
    {
    }
}
```

VISHWAVIDYALAYA INDORE

तमसो मा ज्योतिर्गमय

CLASS WORK
SESSIONAL WORK

ASSIGNMENT / EXPERIMENT No 2

SUBMITTED ON .... MARKS OR GRADE OBTAINED ....

NAME Anushka Tomar .... ROLL NO 19100BTCSEMA05474

CLASS .... DEPARTMENT ....

SUBJECT VR and Arg Reality .... CODE NO ....

Signature of Student

Signature of Professor

Experiment-2

Aim:- Create Single View App using scenekit

program:-

```
import UIkit
import Scenekit
import ARkit

class View Controller : UIView Controller,
        ARSCNView Deligate {
    @IBOutlet var sceneView: ARSCNView!

  override func viewDidLoad() {
   super.viewDidload()
  let configuration = AR World TrackingConfig-
                                        -uration()

   SceneView.session.run(configuration)
  Scene View.delegate = self
   Scene View.showsStatics = true
  let scene = SCNScene(named:"art.scnassets
                                  /ship.scn")!
 }
}
```

Experiment - 3

Aim :- Add Object structure

program :-

```
import UIKit
import Scenekit
import AR kit
class ViewController : UIViewController, ARSCNView
                                    Delegate {

 let configuration = ARWorld Tracking Configuration()
 let node = SCNNode()
 @IBOutlet var sceneView : ARSCNView!
 @IBAction func sphere (_ sender : UIButton) {
     node . removeFromParentNode()
     node . geometry = SCNSphere(radius : 0.1)
     addshape (node : node)
 }

 @IBAction func cone (_ sender : UIButton) {
    node . removeFromParentNode()
    node . geometry = SCNCone(topRadius : 0, bottom
                     Radius : 0.1 , height : 0.1)
    addShape (node : node)

 @IBAction func reset (_ sender : UIButton ) {
    sceneView . Session . pause()
    node . removeFromParentNode()
    sceneView . scene . rootNode . addChildNode(node)
    sceneView . session . run (configuration , options : [.resetTracking])
 }
func addShape(node : SCNNode) {
    node.geometry ? .firstMaterial ? lightingModel=
          .physicallyBased
   node.geometry ? .firstMaterial ? metalness.
    contents = UIColor (white : 1.0, alpha 1.0)
    node . geometry ? . firstMaterial ? diffuse.
    contents = UIColor (white : 0.2, alpha : 1.0)
   node . geometry ? . firstMaterial ? roughness.
   contents = UIColor (white : 0.03, alpha : 1.0)
      node . geometry ? . firstMaterial ? transparencyMode = .dualLayer
   node. geometry ? . firstMaterial . isDoubleSided
          = true

    node . geometry ? . firstMaterial ? . transparency = 0.2

   let scene = SCNScene ()
   scene . rootNode . addChildNode (node)
     sceneView . scene = scene
  }
  override func viewDidLoad ()
  {
    super . viewDidLoad ()
    sceneView . autoenablesDefaultLighting = true
     sceneView . session . run (configuration)
  }
  }
```

Experiment 4

Aim :- Add world origin and feature points

Program :-
```
import UIkit
@main
class AppDeligate : UIResponder , UIApplication-
                                   Deligate {

      func application (_application : UIApplication,
didFinishLaunchingWithOptions launchOptions: [
   UIApplication . LaunchOption key : Any] ? ) -> Bool {
        }

   ~~func~~ return true }
   func application (_ application : UIApplication,
     configurationForConnecting connectingScene Session:
   UI Scene Session , options : UIScene . Connection Op-
   - tions ) -> UI Scene Configuration {
     return UIScene Configuration(name : "Default
   Configuration", sessionRole : connectingSceneSession.rol
}
func application (_ application: UI Application, didDiscard-
   Scene Sessions sceneSessions : Set <UI Scene Session>) {
   }
   }
```

Experiment 5:-

Aim:- Change the position of object use slider in sceneView.

```
program:- import UIkit
          import ARkit
          import Scenekit
          class ViewController : UI ViewController, ARSCNView-
                                                     -Delegate {
    let configuration = ARWorldTracking-
                         Configuration()

    @IBOutlet var sceneView: ARSCNView!
    @IBoutlet var xslider: UISlider!
    @IBoutlet var yslider: UISlider!
    @IBoutlet var zslider: UISlider!
    @IBAction func add Btn(_ sender: UIButton){
      ShowShape() }
    @IBAction func rstBtn(_sender: UIButton) {
      sceneView.session.pause()
    sceneView.scene.rootNode.enumerateChildNodes {
      (node, _) in if node.name == "Sphere" {
            node.removeFromParentNode() {}
    sceneView.session.run(configuration, options:[
      [.resetTracking]) {


func showShape() {
  let node = SCNNode()
  node.geomitry = SCNSphere(radius 0.1)
  node.geometry? firstMaterial?.diffuse.content?
                         = UIColor.orange.
  node.position = SCNVector 3.(xSlider.value,
```

Experiment 6:-

Aim:- Add different shapes in sceneView

program:-

```
        import UIkit
        import ARkit
        import Scenekit
        Class view Controller : UIViewController, ARSCNViewDelegate {
      @ IBOutlet var sceneView : ARSCNView!
  func showShape() {
      let node = SCNNode ()
      node.geometry = SCNSphere(radius : 0.1)
      node.geometry?.firstMaterial?.diffuse.contents
   = UIImage(named : "img.jpeg")
      node position = SCNVector3 (0, 0, 0)
      node.name = "sphere"
      let scene = SCNScene ()
      scene.background.contents = UIImage(named:"img.
                                                  jpg")
      scene View.scene = scene
        Scene.rootNode.add Child Node(node) {
  override func view DidLoad () {
        super.viewDidLoad ()
        sceneView.delegate = self
        SceneView.showsStatics = true
        sceneView.debug Option = [ARSCNDebugOption.
        ShowWorld Origin, ARSCNDebugOption.show-
            -feature Points]
        ShowShap ()
        }
      }
```

Experiment 7 :-

Aim

program :-
```
import UIkit
import ARkit
import Scene kit
class View Controller : UIView Controller, ARSCN
@IB Outlet var sceneView: ARSCNView!      -View Delegate {
let config = ARWorld Tracking Configuration ()
func ShowShape () {
  let node = SCNNode ()
  node . geometry = SCNSphere (radius: 0.1 )
  node : geometry ? first Material ? diffuse . contents =
     UIColor . orange
  node . position = SCNVector 3 (0, 0, 0)
  node . name = "sphere"
  let scene = SCNScene ()
  SceneView . scene = scene ()
  scene . root Node . add Child Node (node) }
@IB Action func reset (_ sender : Any ) { sceneView . session
 pause ()      sceneView . session . run (config, option: [ . reset Tracking])
{ Override func viewDidLoad () }
    Super . viewDidLoad ()
    SceneView . delegate = self
    SceneView . ShowStatistics = true
    SceneView . autoenables Default Lighting = true
  sceneView . debug Options = [ARSCNDebugOption . show-
 WorldOrigin, ARSCNDebugOption . showFeature Points]
    show shape ()
 }}
```