Program 1

class Test{

    final int x;

    final int y;

    const Test(this.x,this.y){

        print("In const constructor");

    }

}

void main(){

    Test obj= Test(10,20);

    print(obj.x);

}

**Error:-**A const constructor can't have a body. Try removing either the 'const' keyword or the body. const Test(this.x,this.y){ ^^^^^

**Explanation:-**A constant Keyword cant have a body

**Program:-2**

class Employee {

  int? empId;

  String? empName;

  Employee() {}

  Employee(int empId, String empName) {}

}

void main() {

  Employee obj = new Employee();

}

**Error**: 'Employee' is already declared in this scope. Employee(int empId, String empName) {} ^^^^^^^^ ../../../../Desktop/DART/6-1-24/dart2.dart:5:3: Context: Previous declaration of 'Employee'. Employee() {} ^^^^^^^^ ../../../../Desktop/DART/6-1-24/dart2.dart:10:22: Error: Can't use 'Employee' because it is declared more than once. Employee obj = new Employee(); ^

Explanation:- two constructors with same name can’t be use in one class

**Program:-3**

class Demo {

  final int? x;

  final String? str;

  const Demo(this.x, this.str);

}

void main() {

  Demo obj1 = const Demo(10, "Core2web");

  print(obj1.hashCode);

  Demo obj2 = const Demo(10, "Biencaps");

  print(obj2.hashCode);

}

**O/P:- 774754847**

**1063770429**

**Explanation:-** 2Objects are created with different parameters that’s why its pointing to the different objects which give the different addresses

**Program:-4**

class Company {

  int empcount;

  String compName;

  Company(this.empcount, [this.compName = "Biencaps"]);

  void compInfo() {

    print(empcount);

    print(compName);

  }

}

void main() {

  Company obj1 = new Company(100);

  Company obj2 = new Company(200, "Pubmatic");

  obj1.compInfo();

  obj2.compInfo();

}

**O/P:-** 100

Biencaps

200

Pubmatic

**Explanation:-** While the parameters one parameter compName has been already default set as “biencaps”, which can be overridden while creating obj or if not given then the default is being used

**Program 5:-**

class Company {

  int? x;

  String? str;

  Company(this.x, {this.str = "Core2web"});

  void compInfo() {

    print(x);

    print(str);

  }

}

void main() {

  Company obj1 = new Company(100);

  Company obj2 = new Company(200, "Incubator");

  obj1.compInfo();

  obj2.compInfo();

}

**Error: Too many positional arguments: 1 allowed, but 2 found. Try removing the extra positional arguments. Company obj2 = new Company(200, "Incubator"); ^ ../../../../Desktop/DART/6-1-24/dart5.dart:5:3: Context: Found this candidate, but the arguments don't match. Company(this.x, {this.str = "Core2web"}); ^^^^^^^**

**Explanation:-** Here we are trying to set default parameter in curly braces which is not allowed that’s why its giving error

**Program:-6**

class Company {

  int? empcount;

  String compName;

  Company({this.empcount, this.compName = "Deloitte"});

  void compInfo() {

    print(empcount);

    print(compName);

  }

}

void main() {

  Company obj1 = new Company(empcount: 100, compName: "Veritas");

  Company obj2 = new Company(compName: "Pumatic", empcount: 200);

  obj1.compInfo();

  obj2.compInfo();

}

**O/P:-100**

**Veritas**

**200**

**Pumatic**

**Explanation:-** here the parameterized constructor has been to get the proper parameters which creating the object

**Program 7:-**

class Point {

  int x;

  int y;

}

void main() {

  Point obj = Point();

}

Error: Field 'x' should be initialized because its type 'int' doesn't allow null. int x; ^ ../../../../Desktop/DART/6-1-24/dart7.dart:3:7: Error: Field 'y' should be initialized because its type 'int' doesn't allow null. int y; ^

**Explanation:-** Here we are declaring variables but not initializing it that why its giving error

**Program 8:-**

class player{

  int? jerNo;

  String? pName;

  const Player(this.jerNo,this.pName);

}

 void main(){

  Player obj =const(45,"Rohit");

 }

**Error: Getters, setters and methods can't be declared to be 'const'. Try removing the 'const' keyword. const Player(this.jerNo,this.pName); ^^^^^ ../../../../Desktop/DART/6-1-24/dart8.dart:1:7: Error: The non-abstract class 'player' is missing implementations for these members: - player.Player Try to either - provide an implementation, - inherit an implementation from a superclass or mixin, - mark the class as abstract, or - provide a 'noSuchMethod' implementation. class player{ ^^^^^^ ../../../../Desktop/DART/6-1-24/dart8.dart:5:9: Context: 'player.Player' is defined here. const Player(this.jerNo,this.pName); ^^^^^^ ../../../../Desktop/DART/6-1-24/dart8.dart:5:16: Error: Field formal parameters can only be used in a constructor. Try removing 'this.'. const Player(this.jerNo,this.pName); ^^^^ ../../../../Desktop/DART/6-1-24/dart8.dart:5:27: Error: Field formal parameters can only be used in a constructor. Try removing 'this.'. const Player(this.jerNo,this.pName); ^^^^ ../../../../Desktop/DART/6-1-24/dart8.dart:9:3: Error: 'Player' isn't a type. Player obj =const(45,"Rohit"); ^^^^^^**

**Explanation:-** While creating obj we are not passing values rather assigning it that’s why its giving error

**Program 9:-**

int a = 10;

class Test {

  int x = 20;

  int y = 20;

  static num z = 30;

  Test(this.x, this.y, this.a);

  void fun() {

    print(a);

    print(x);

    print(y);

  }

}

void main() {

  Test obj = Test(10, 30, 40);

  obj.fun();

}

**Error: 'a' isn't an instance field of this class. Test(this.x, this.y, this.a); ^**

**Explanation:-** a is not a variable of class test that’s why its giving error

**Program 10:-**

class Demo {

  Demo() {

    print("In Demo");

  }

  factory Demo() {

    print("In demo factory");

    return Demo();

  }

}

void main() {

  Demo obj = new Demo();

}

**Error: 'Demo' is already declared in this scope. factory Demo() { ^^^^ ../../../../Desktop/DART/6-1-24/dart10.dart:2:3: Context: Previous declaration of 'Demo'. Demo() { ^^^^ ../../../../Desktop/DART/6-1-24/dart10.dart:7:12: Error: Can't use 'Demo' because it is declared more than once. return Demo(); ^ ../../../../Desktop/DART/6-1-24/dart10.dart:12:18: Error: Can't use 'Demo' because it is declared more than once. Demo obj = new Demo(); ^**

**Explanation:-** In same class we cannot declare constructor with same name more than once

**Program:-11**

class Test {

  Test.\_private() {

    print("In demo");

  }

  factory Test() {

    print("In demo factory");

    return Test.\_private();

  }

}

void main() {

  Test obj = new Test();

}

**O/P:- In demo**

**factory In demo**

**Explanation:-** factory const can returns its object