**TASK1**

**DART PRACTICE**

|  |
| --- |
| void main() {  print('Hello World!');  } |

**Variables and Data Types:**

|  |
| --- |
| void main() {  int age = 25;  double height = 5.3;  String name = 'Hafsa’ ;  bool check = true;    print('Age: $age');  print('Height: $height');  print('Name: $name');  print('check: $check’);    } |

**Control Flow Statements**

**If-Else Statement**

|  |
| --- |
| void main() {  int number = 10;  if (number > 0) {  print('$number is positive.');  } else if (number < 0) {  print('$number is negative.');  } else {  print('$number is zero.');  }  } |

**Switch-Case Statement:**

|  |
| --- |
| void main() {  String grade = 'A';  switch (grade) {  case 'A':  print('Excellent');  break;  case 'B':  print('Good');  break;  case 'C':  print('Fair');  break;  case 'D':  print('Poor');  break;  default:  print('Invalid grade');  }  } |

**Loops:**

|  |
| --- |
| void main() {  for (int i = 1; i <= 5; i++) {  print('Number: $i');  }  } |

**While Loop:**

|  |
| --- |
| void main() {  int i = 1;  while (i <= 5) {  print('Number: $i');  i++;  }  } |

**Do while loop:**

|  |
| --- |
| void main() {  int i = 1;  do {  print('Number: $i');  i++;  } while (i <= 5);  } |

**Function:**

|  |
| --- |
| void main() {  int result = add(4, 5);  print('Result: $result');  }  int add(int a, int b) {  return a + b;  } |

**OOP**

|  |
| --- |
| class A {  int a = 1;  int b = 2;  A(this.a, this.b);  void display() {  print('Number: $a');  print('Number: $b');  }  }  void main() {  A obj1 = new A(2, 4);  obj1.display();  } |

**Inheritance:**

|  |
| --- |
| class Animal {  void eat() {  print('Eating...');  }  }  class Dog extends Animal {  void bark() {  print('Barking...');  }  }  void main() {  Dog dog = Dog();  dog.eat();  dog.bark();  } |

**Polymorphism:**

|  |
| --- |
| class Animal {  void makeSound() {  print('Animal sound');  }  }  class Dog extends Animal {  @override  void makeSound() {  print('Bark');  }  }  class Cat extends Animal {  @override  void makeSound() {  print('Meow');  }  }  void main() {  Animal myDog = Dog();  myDog.makeSound(); // Bark  Animal myCat = Cat();  myCat.makeSound(); // Meow  } |

**Abstract Classes and Interfaces:**

|  |
| --- |
| abstract class Shape {  void draw(); // Abstract method  }  class Circle extends Shape {  @override  void draw() {  print('Drawing Circle');  }  }  class Square extends Shape {  @override  void draw() {  print('Drawing Square');  }  }  void main() {  Shape circle = Circle();  circle.draw(); // Drawing Circle  Shape square = Square();  square.draw(); // Drawing Square  } |

**Encapsulation:**

|  |
| --- |
| class Rectangle {  double \_width;  double \_height;  Rectangle(this.\_width, this.\_height);  double get width => \_width;  set width(double value) {  if (value > 0) {  \_width = value;  }  }  double get height => \_height;  set height(double value) {  if (value > 0) {  \_height = value;  }  }  double get area => \_width \* \_height;  }  void main() {  Rectangle rect = Rectangle(5, 10);  print('Area: ${rect.area}');  rect.width = 7;  print('Updated Area: ${rect.area}');  } |