

Start with the following list of numbers:

```
const numbers = [1, 2, 4, 7];
```

Print the square of each number: 1, 4, 16 and 49.

1. First, use a for loop.
2. Solve the problem again using a for-in loop.

Exercise 2:

1. Create a class named User as follows:

```
class User {  
  int id = 213;  
  String name = 'Fatima';  
}
```

2. Create an object using this class
3. Print the object you created.
4. What will appear in the screen.
5. Define a toString method.
6. Print the object again
7. What will appear in the screen.

Exercise 3:

1. Create a constructor using the long-form, Short-Form, name constructor, and forwarding methods to initialize the id and the name of the user
2. Use the constructors to create an object from the class
8. Define a toString method.
9. Print the object again for each constructor you used

```
10. class User {  
11.   int id=6;  
12.   String name='a';  
13.   User (int id,String name){  
14.     this.id=id;  
15.     this.name=name;  
16.  
17.   }  
18.   User.shortform(this.id,this.name);  
19.   User.namedConstructor( {int id=0,String name='Unknown'}){  
20.     this.id=id;  
21.     this.name=name;  
22.   }  
23.   User.forwardingConstructor(String name):
```

```

24. this(0,name);
25. @override
26. String toString(){
27.   return'User{id:$id,name:$name}';
28. }
29.}
30.
31.void main() {
32.User user1=User(123,'john');
33. print(user1);
34. User user2=User.shortform(456,'jahn');
35. print(user2);
36. User user3=User.namedConstructor(name:'Bob');
37. print(user3);
38. User user4=User.forwardingConstructor('alice');
39. print(user4);
40. }

```

The screenshot shows the DartPad web interface. The code editor on the left contains the Dart code from the previous block. The console on the right shows the output of the code, which is the string representation of the four User objects created in the main function. The output is:

```

User{id:123, name:john}
User{id:456, name:jahn}
User{id:0, name:Bob}
User{id:0, name:alice}

```

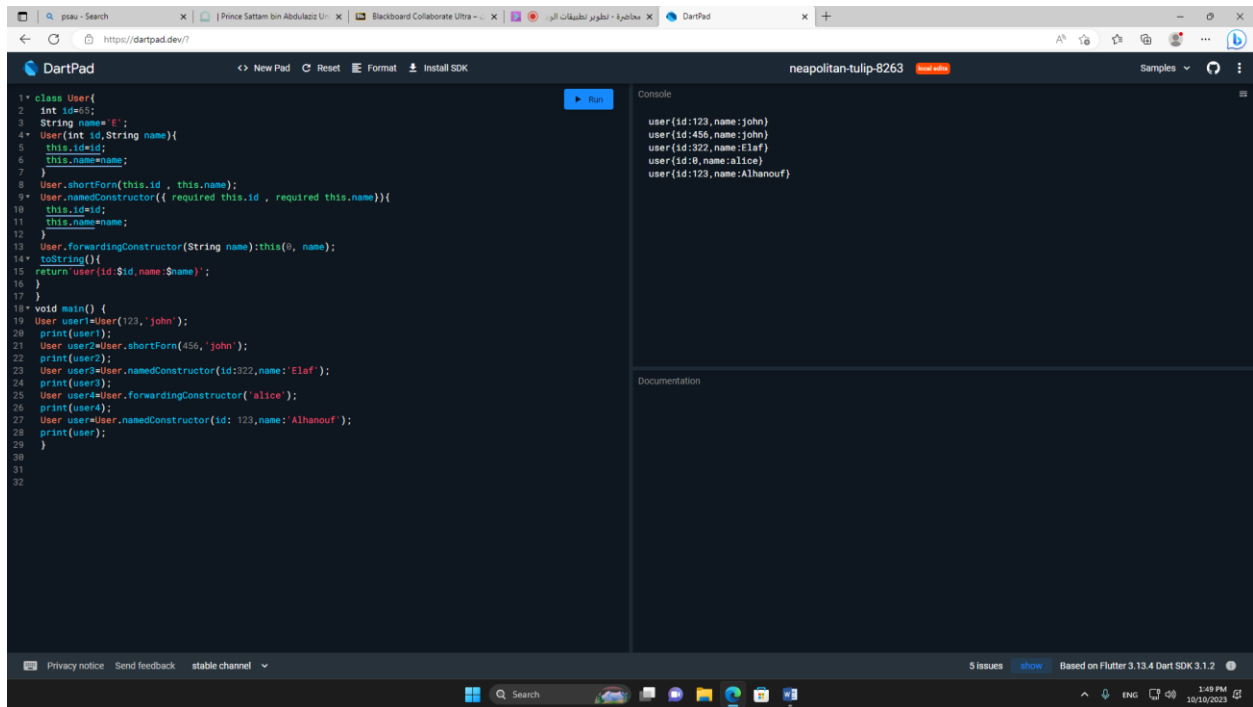
The interface also shows the DartPad logo, a 'Run' button, and a 'Documentation' link. The bottom of the interface shows the Flutter 3.13.4 Dart SDK 3.1.2 version and the time 1:32 PM on 10/10/2023.

41.

1. Define one of the previous constructor with name parameters.
2. Create object and print it.

```
class User{
    int id=65;
    String name='E';
    User(int id,String name){
        this.id=id;
        this.name=name;
    }
    User.shortForn(this.id , this.name);
    User.namedConstructor({ required this.id , required this.name}){
        this.id=id;
        this.name=name;
    }
    User.forwardingConstructor(String name):this(0, name);
    toString(){
return'user{id:$id,name:$name}';
    }
}

void main() {
    User user1=User(123,'john');
    print(user1);
    User user2=User.shortForn(456,'john');
    print(user2);
    User user3=User.namedConstructor(id:322,name:'Elaf');
    print(user3);
    User user4=User.forwardingConstructor('alice');
    print(user4);
    User user=User.namedConstructor(id: 123,name:'Alhanouf');
    print(user);
}
```



Exercise 5:

1. Define the previous class with private instance variables
2. Use set and get methods

```

class UserPrivate{
  int id=65;
  String name='E';
  UserPrivate(this.id,this.name);
  int get id1=>id;
  set id1(int v){
    id=v;
  }
  String get name1=>name;
  set name1(String){
    name='v';
  }
  String toString(){
    return'user {id:$id,string:$name}';
  }
}

```

```

void main() {
UserPrivate user1= UserPrivate(123,'Alhanouf');
print(user1);
user1.id=683;
user1.name="Elaf";
print(user1);

}

```

