Bilkent University

Department of Computer Engineering

CS 315

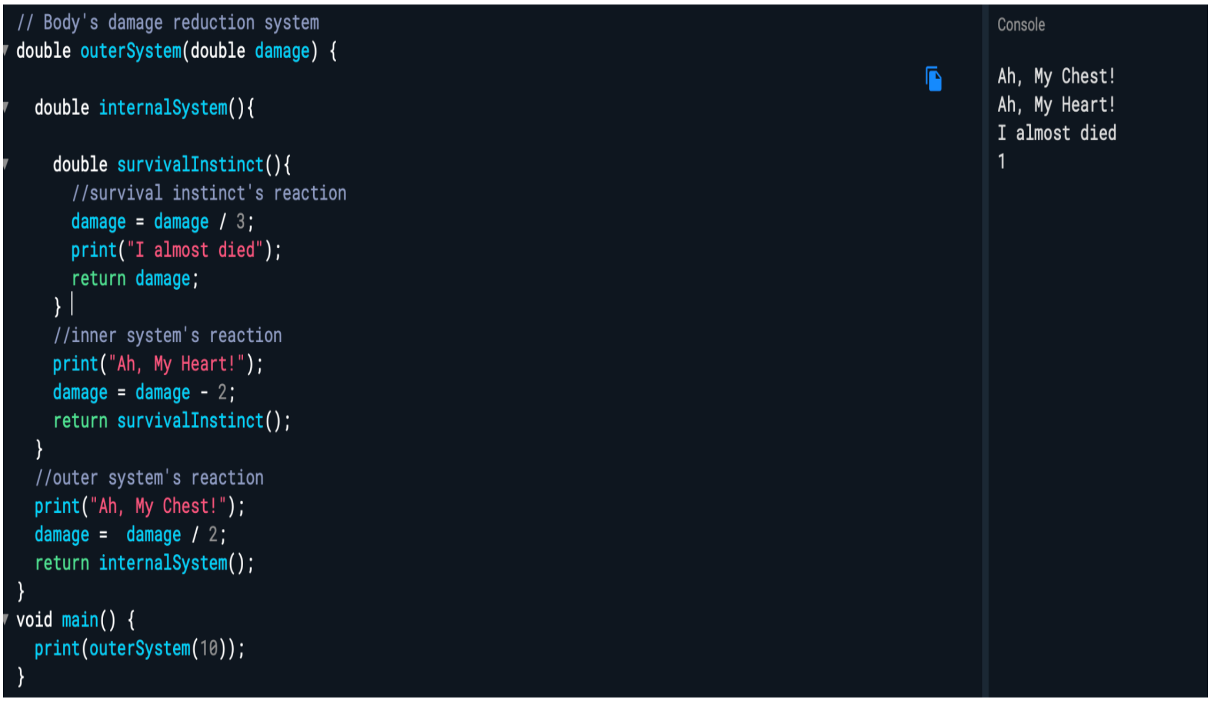
Homework Assignment 3

Subprograms in Dart

Alper Bozkurt
21802766

1 - Nested subprogram definitions

In Dart Language, nested subprograms could be defined as in the example code. Code is about the reduction system of the body. The outer system takes damage (10) as a parameter. After reducing damage to half (5), it returns the subfunction whose name is Internal System. There is no need to pass the damage variable to that subfunction because it is in the inner function. Thus, in the inner function reduced damaged variable can be used. Internal System reduces the damage by two (3) and in the end, return “survivalInstinct”. It divides the damage by 3 and return damage. In the Main method the returned damage value is printed (1). To understand the order, the systems talk. Outer System prints “my chest”, Internal System prints “my heart”, and Survival System prints “I almost died”.

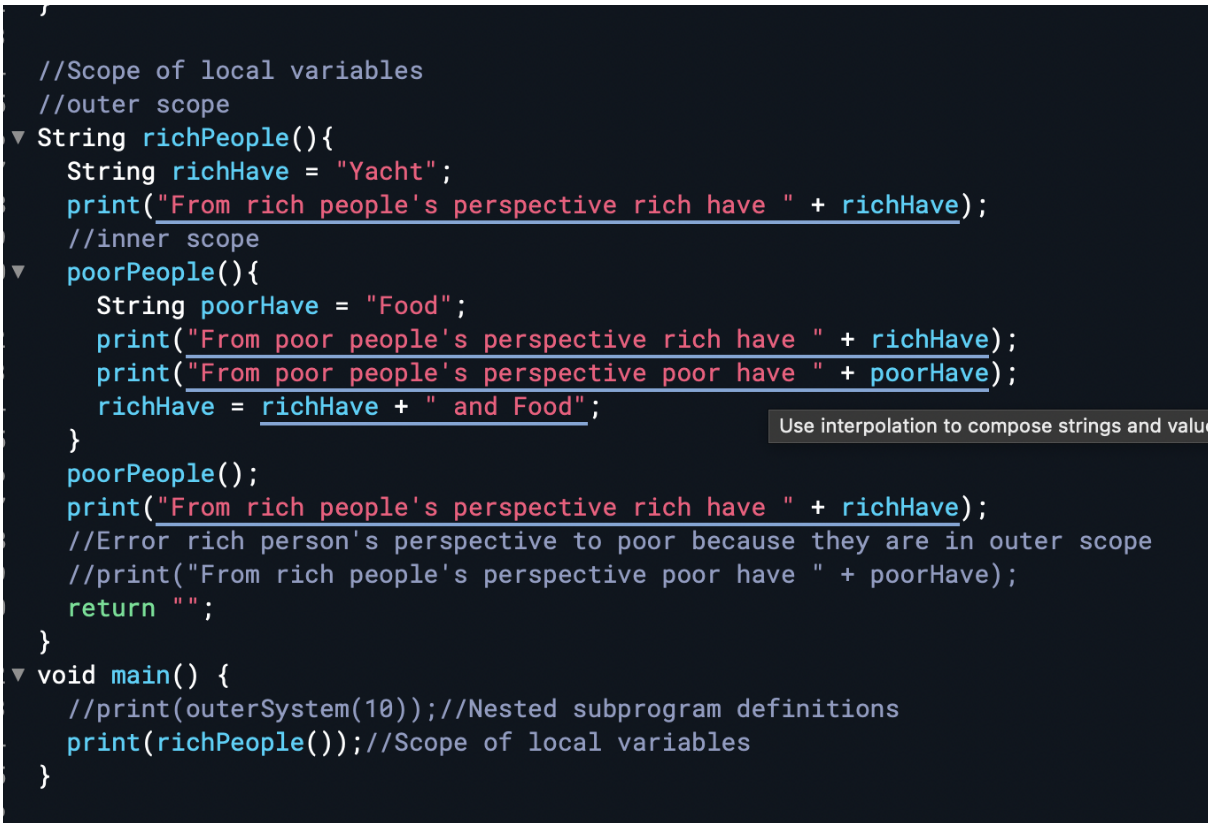


```
// Body's damage reduction system
double outerSystem(double damage) {

  double internalSystem(){

    double survivalInstinct(){
      //survival instinct's reaction
      damage = damage / 3;
      print("I almost died");
      return damage;
    }
    //inner system's reaction
    print("Ah, My Heart!");
    damage = damage - 2;
    return survivalInstinct();
  }
  //outer system's reaction
  print("Ah, My Chest!");
  damage =  damage / 2;
  return internalSystem();
}
void main() {
  print(outerSystem(10));
}
```

```
Ah, My Chest!
Ah, My Heart!
I almost died
1
```

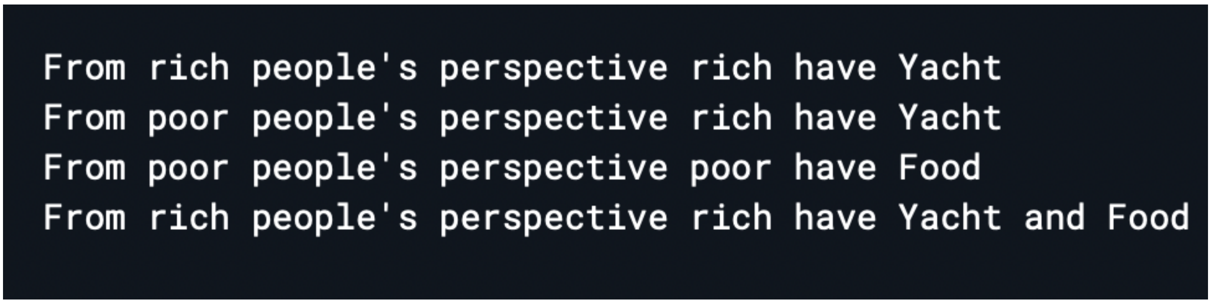
2 - Scope of local variables

In Dart Language, the scope of local variables could apply as in the example code. The code is about the perspectives of people with different income levels. The outer scope is to define rich perspective, and the variable is to demonstrate what they have. The first illustration is to prove that outer scope variable displays in outer scope in the same way. In that code example, it is explained as from rich people's perspective rich have "Yacht" which is an outer scope variable. There is an inner scope called poorPeople, which is to symbolize poor people's perspective. Inner scope has "Food" for the variable. In the second illustration, outer scope variable is displayed in the inner scope, which means, from poor people's perspective, what rich people have. In the third illustration, the inner scope variable is displayed in the inner scope, which means, from poor people's perspective, what poor people have. In the fourth illustration, the outer scope variable is manipulated in the inner scope. Rich people have also "Food". Because Dart is a dynamically typed language. Manipulation in the poor people's perspective affects the general fact perspective about what rich people. For the fifth illustration, the rich people's perspective of what poor people have is investigated, but an error occurs. It is proven that from the outer scope, there is no reach of a variable that is declared in the inner scope. After all the illustrations, "return " "" used for not getting a null print after the code is executed.



```
}

//Scope of local variables
//outer scope
String richPeople(){
  String richHave = "Yacht";
  print("From rich people's perspective rich have " + richHave);
  //inner scope
  poorPeople(){
    String poorHave = "Food";
    print("From poor people's perspective rich have " + richHave);
    print("From poor people's perspective poor have " + poorHave);
    richHave = richHave + " and Food";
  }
  poorPeople();
  print("From rich people's perspective rich have " + richHave);
  //Error rich person's perspective to poor because they are in outer scope
  //print("From rich people's perspective poor have " + poorHave);
  return "";
}
void main() {
  //print(outerSystem(10));//Nested subprogram definitions
  print(richPeople());//Scope of local variables
}
```

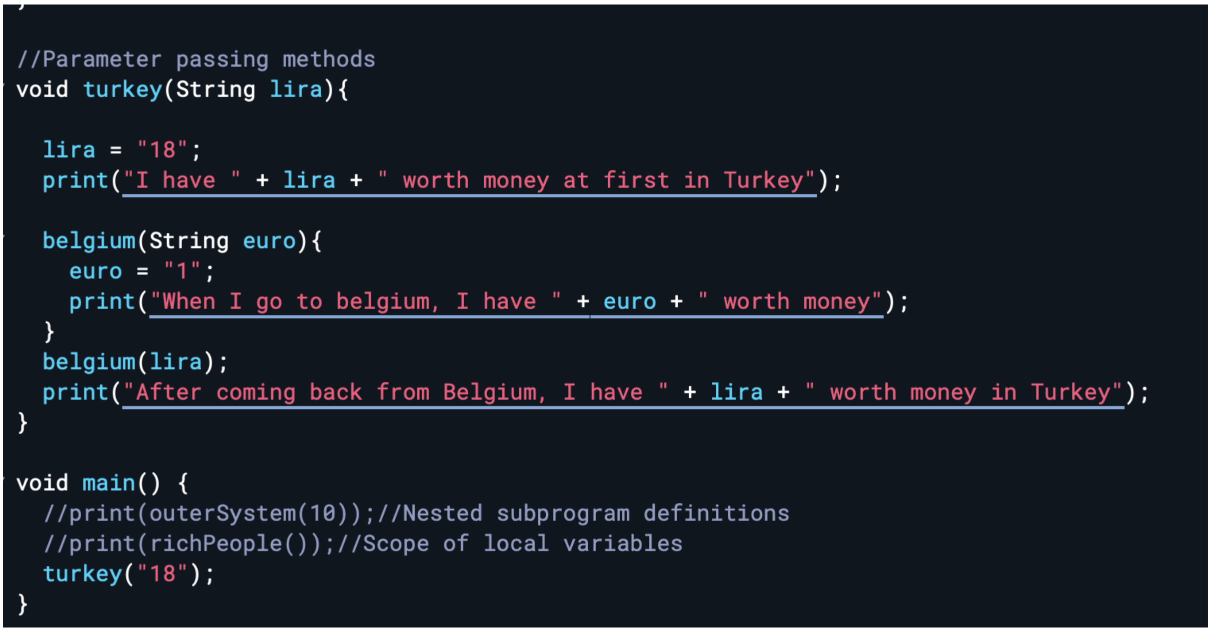
Use interpolation to compose strings and valu



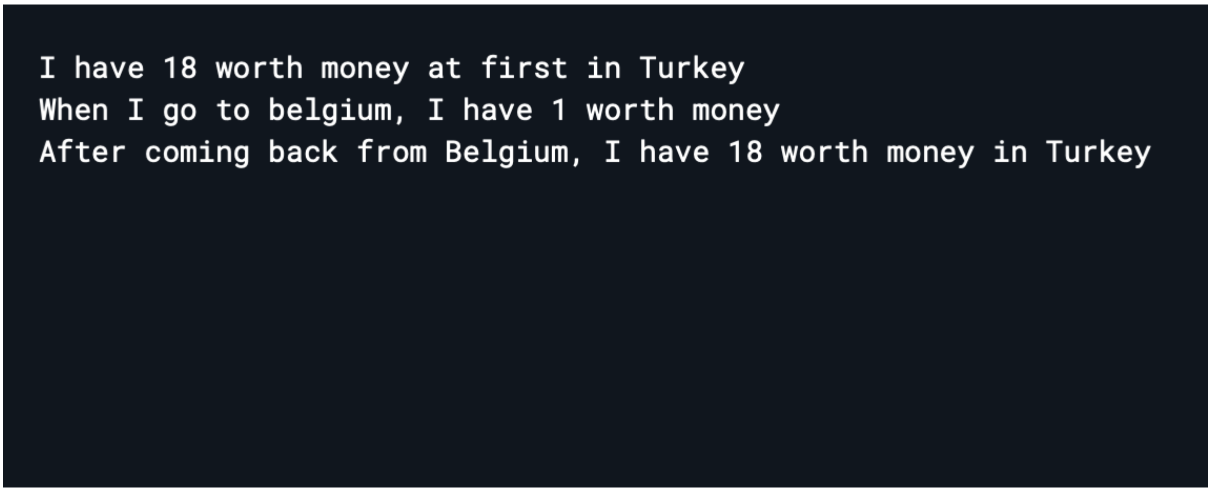
```
From rich people's perspective rich have Yacht
From poor people's perspective rich have Yacht
From poor people's perspective poor have Food
From rich people's perspective rich have Yacht and Food
```

3 - Parameter passing methods

In Dart Language, parameter pass in methods could be done as in the example code. The code is about understanding the value for money difference between Belgium and Turkey. The outer scope is Turkey takes a lira parameter which is symbolize the initial money haven at first. After display the initial value, if it is gone to Belgium with the same money, the value of money changed from “18” to “1”. Then, in the last illustration, it is the situation after coming back to outer scope (“turkey”). After displaying the value of the same money, it illustrates “18” again. The eccentric part is that if belgium method not taking parameter the value of money will not change after coming back to Turkey. When declared a parameter, the new variable assigned in that “belgium” function which is euro in that example. Euro become different money currency from lira. Moreover, as it is stated in the scope part, lira can be accessible in belgium method, however, turkey does not recognize the euro as currency.



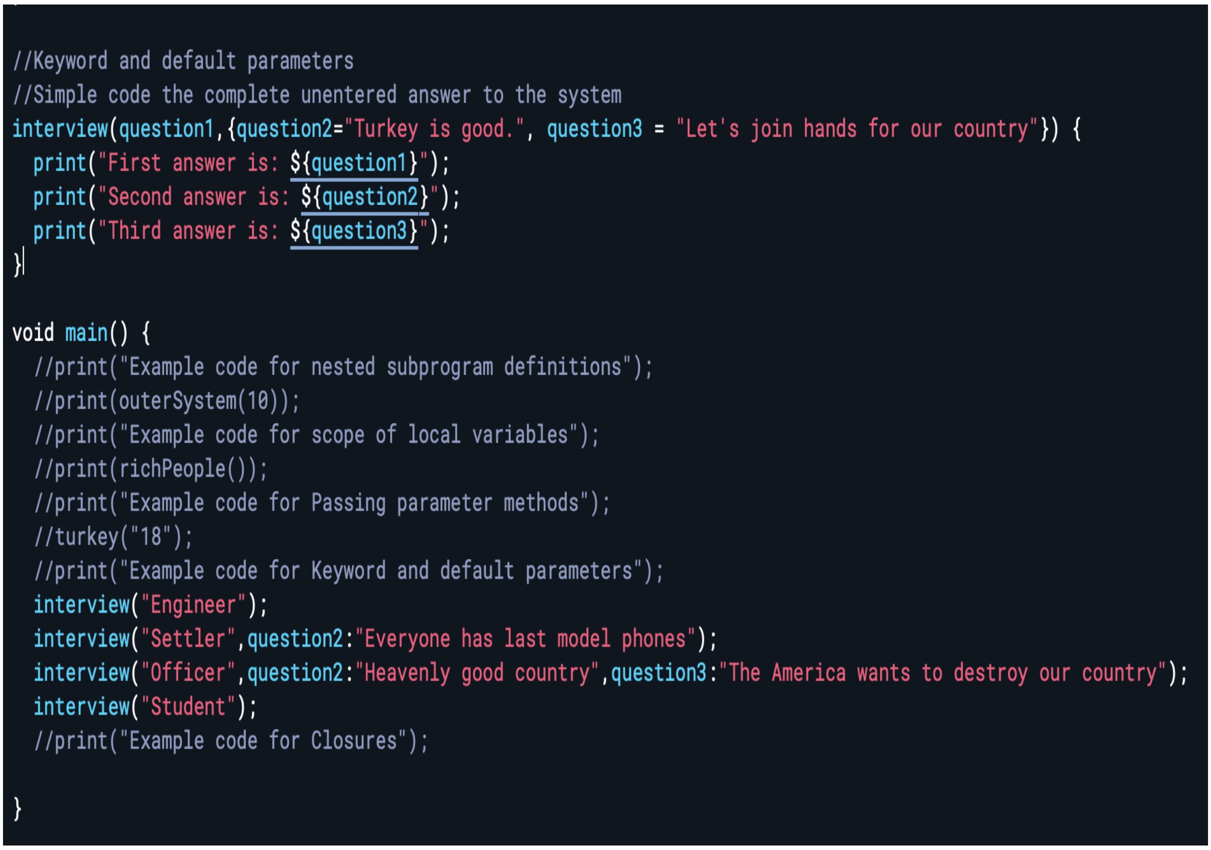
```
//Parameter passing methods
void turkey(String lira){

  lira = "18";
  print("I have " + lira + " worth money at first in Turkey");

  belgium(String euro){
    euro = "1";
    print("When I go to belgium, I have " + euro + " worth money");
  }
  belgium(lira);
  print("After coming back from Belgium, I have " + lira + " worth money in Turkey");
}

void main() {
  //print(outerSystem(10));//Nested subprogram definitions
  //print(richPeople());//Scope of local variables
  turkey("18");
}
```



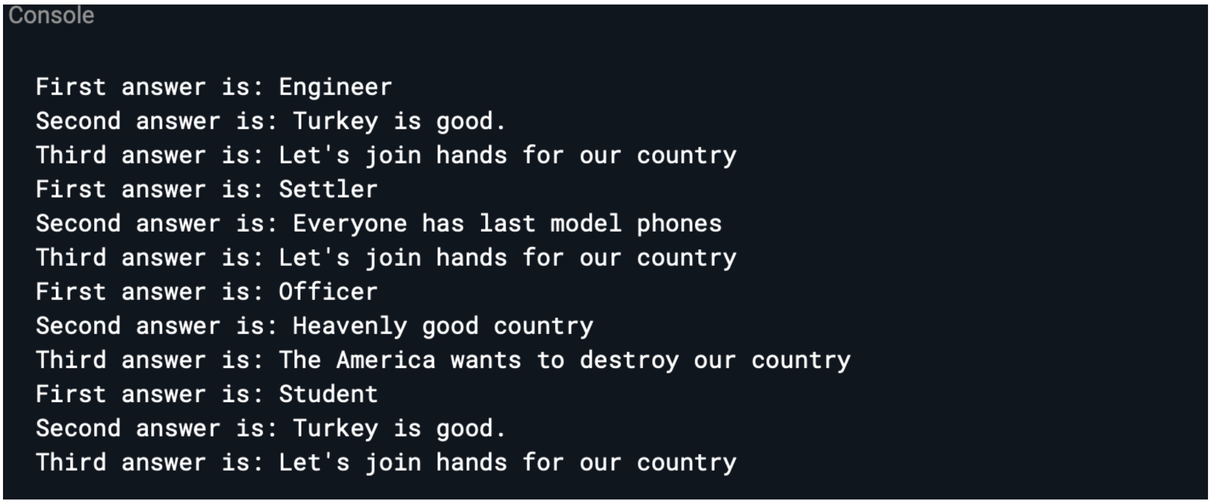
```
I have 18 worth money at first in Turkey
When I go to belgium, I have 1 worth money
After coming back from Belgium, I have 18 worth money in Turkey
```

4 - Keyword and default parameters

In Dart Language, keyword and default parameters could apply as in the example code. The code makes censorship after interviews with different people. The program fills the blank answers automatically. Default parameters are used for blank answers. There are four people who have interviews. Engineer and student have no answer in the system for the first and second questions, settler has no answer for the third question and the officer answered all the questions. For non-answered question program fill in the blanks with desired answers. For the keyword parameter, it holds the value by a specified keyword which is the question1 in that situation. All people who answer question1 are passed to the interview function.



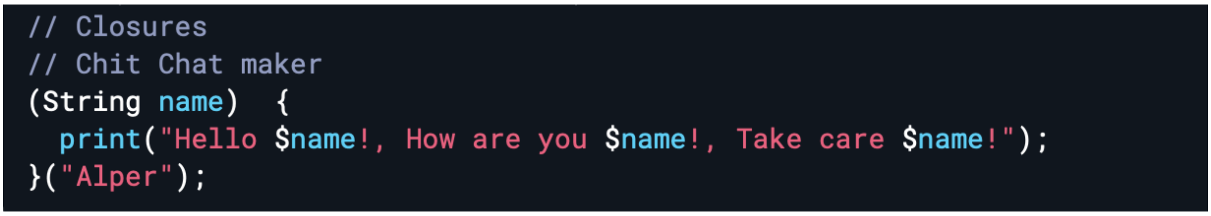
```
//Keyword and default parameters
//Simple code the complete unentered answer to the system
interview(question1,{question2="Turkey is good.", question3 = "Let's join hands for our country"}) {
  print("First answer is: ${question1}");
  print("Second answer is: ${question2}");
  print("Third answer is: ${question3}");
}

void main() {
  //print("Example code for nested subprogram definitions");
  //print(outerSystem(10));
  //print("Example code for scope of local variables");
  //print(richPeople());
  //print("Example code for Passing parameter methods");
  //turkey("18");
  //print("Example code for Keyword and default parameters");
  interview("Engineer");
  interview("Settler",question2:"Everyone has last model phones");
  interview("Officer",question2:"Heavenly good country",question3:"The America wants to destroy our country");
  interview("Student");
  //print("Example code for Closures");

}
```



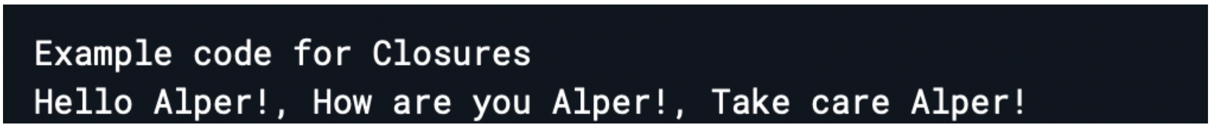
```
First answer is: Engineer
Second answer is: Turkey is good.
Third answer is: Let's join hands for our country
First answer is: Settler
Second answer is: Everyone has last model phones
Third answer is: Let's join hands for our country
First answer is: Officer
Second answer is: Heavenly good country
Third answer is: The America wants to destroy our country
First answer is: Student
Second answer is: Turkey is good.
Third answer is: Let's join hands for our country
```

5 - Closures

In Dart Language, closures could make as in the example code. The code, creates the simple chit chat with the given name. Closure is made by brackets, inside the brackets type and variable name is entered. After the operation between curly brackets, the wanted assigned value is written, in that example it is the name of the intended person.



```
// Closures
// Chit Chat maker
(String name)  {
  print("Hello $name!, How are you $name!, Take care $name!");
}("Alper");
```



```
Example code for Closures
Hello Alper!, How are you Alper!, Take care Alper!
```

6 - Evaluation of Languages

Dart is a highly readable programming language of subprogram syntax. Functions are declared straightforward. The syntax is similar to C++ which is very understandable. The statements are in curly brackets and have an indentation. It can be easily followed the scope and differentiate it. It is also a very writable programming language of subprogram syntax. It has a simple syntax, simple methods and function declaration. To sum up, Dart is a pretty writable and readable language.

7 - Learning Strategy

Dart is my favorite language from past homework. While learning it there is also no problem for me. There are options to express the same situation in different ways. The hardest part becomes understanding what the homework wanted. I also spend significant time performing what I understand in a creative and simple way. The outcome is quite satisfying for me. The motivation for the effort comes from if I totally understand the concept and language, moreover, if I have the capacity to write what I wanted to write. There is much need the external sources to satisfy the homework because I adapt the syntax and there is no need to know highly complicated information about the Dart language. Searching for what I want to know becomes easier after the previous three assignments. I write the code in the dartpad.(https://dartpad.dev/?)

https://www.educative.io/answers/how-to-create-a-nested-function-in-dart

https://www.educative.io/answers/what-is-a-dynamic-type-in-dart

https://medium.com/android-news/magic-lies-here-statically-typed-vs-dynamically-typed-languages-d151c7f95e2b

https://stackoverflow.com/questions/43334714/pass-a-typed-function-as-a-parameter-in-dart https://www.w3adda.com/dart-tutorial/dart-default-parameter-values

https://o7planning.org/14061/dart-closures