Department of Computer Engineering

BLG 252E
Object Oriented Programming

III. Homework Report

ID	Name	Surname
150150043	Yavuz	Koca

Homework Assistant: Muhammet Şahin

1 INTRODUCTION

In this homework, I have learned how to use templates and exceptions properly.

1.1 COMPILE

To compile and run this program you can write this command in your terminal

```
g++ *.cpp –std=c++14
```

Then run the program with `./a.out`

2 .GENERIC ARRAY CLASS

In pdf, some attributes has set as public but I think its not very safe method. That's why I changed some of it private. Also I have write some extra functions which will be explained below.

2.1 ATTRIBUTES

size

Size attribute holds the size of the elements array. It is set to constant because it will not change after the constructor has called.

elements

Elements attribute's type is T, because GenericArray can hold more than one type of classes i.e. int, double, Money etc.

total

This attribute holds the total amount of elements array. Its type is T because it is depends on GenericArray type.

2.2 METHODS

checkBonusOrPenaltySituation

This functions checks the total attribute. If it is more than 100, adds 5 more. Else, subtracts 5.

GenericArray

This constructor first takes memory for elements pointer with given size. Then sets all objects in the elements array to zero to prevent garbage values.

push

This function takes two argument. First one is the objects itself (int, double or Money), second one is the index number to put the object in the elements array. If object is less than zero or index number is not appropriate, then throw an exception.

print
Prints the objects inside of the elements array.

sum
With in the for loop, adds all objects in the elements array and writes it into total attribute. Then checkBonusOrPenaltySituation function is called to check the total attribute's value to add bonus or subtract penalty.

~GenericArray
This destructor, frees the taken memory in the constructor.

3 MONEY

3.1 ATTRIBUTES

lira
Holds the amount of lira

kurus
Holds the amount of kurus

3.2 METHODS

Money(int, int)
This constructor checks if the lira or kurus is below than zero or not. If it is, throw an exception.

Money() = default
When we write "this->elements = new T[size]", we call Money() = default constructor. That's why I initialize it but its empty. I could have set the all objects in the array to zero in this function but I wrote it in the GenericArray's constructor. So it's the same thing.

operator=(int)
When I want to say Money a = 0; I needed this operator overloading. Because I needed to say: this->total = 0 in some cases i.e. in sum function.

operator<(int)
It checks if the lira or kurus is less than given integer number. I write it because if it is less than zero, throws an exception. Its constant because no attribute has changed.

operator>(int)
It checks if the lira attribute is more than given integer or not. I write it to call this in checkBonusOrPenaltySituation function. If it is more than 100, bonus added. Its constant because no attribute has changed.

operator +=(const Money&money)
This function has written for the sum function in the GenericArray class. Adds lira and kurus attributes. At the end, if kurus is more than 100, add the carry amount to lira attribute and update the kurus attribute.

operator+=(int)
If money object is more than 100 lira, add 5 more lira to lira attribute. That’s why I wrote this function.

operator-=(int)
If money object is less than 100 lira, subtract 5 lira from lira attribute

operator<<(ostream&, const Money)
to print current money by writing "std::cout << this->total;" I wrote this function.