**FPU Fall 2020 – OOP COP 3337C Homework-4**

**Due Date & Time**: **11/30/2020** – 11.59pm through CANVAS.

1) **Subject: Polymorphism**

Create one .cpp file for the following question and upload it under homework-4 in the Canvas. Use Your First Name + Your Last Name + Part1 + HW4. For example, Bayazit_Karaman_Part1_HW4.cpp.

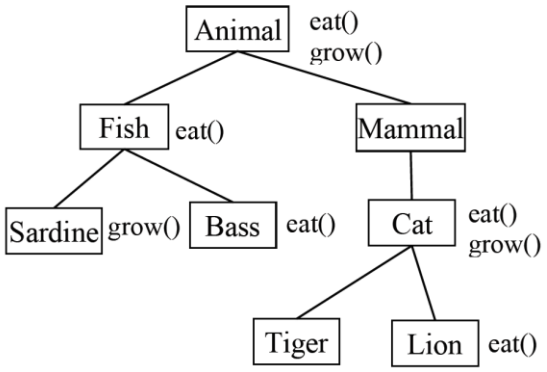


Figure - 1

Write a complete C++ program (ONE FILE) and create 8 classes that form a class tree as shown in the Figure-1. There is no need to split the code due to the simplicity of all classes. Every class has only one no-argument constructor and no data field. Each class may have an eat() method and/or a grow() method. Both functions have no return and no parameter. Each no-argument constructor should print "constructor of xxx class" where "xxx" represents the name of the current class. Each eat() method or each grow() method, should print "eat() method of xxx class" or "grow() method of xxx class", respectively.

Add another function above the main() function with given code below.

```
void display(Animal& a1) {
   a1.eat();
   a1.grow();
}
```

In the main() function, create one object for each leaf classes, i.e. classes Sardine, Bass, Tiger, and Lion. Then call both eat() and grow() functions from each object.

2) Create a C++ program and upload it under Homework-4 in the Canvas. Use Your First Name + Your Last Name + Part-1 + HW4. For example, Bayazit_Karaman_Part2_HW4.cpp.

In this homework, you will implement the Time class with overloaded operators. Two files in Module Week-14 are given for this assignment (Time.h, testTime.cpp). You should create a new file Time.cpp and add implementation of all methods listed in Time.h:

```
Time(unsigned int h = 0, unsigned int m = 0);
```
User defined constructor initializing class data fields with a given values. If the given number of minutes is greater than 59, constructor should calculate the appropriate number of minutes and hours.

```
void showTime();
```
Display current value of hours and minutes.

```
Time operator+ (const Time&) const;
```
Overloaded operator + for adding two times.
*1h 10min + 1h 15min = 2h 25min.*

```
bool operator> (const Time&) const;
```
Overloaded operator > for comparing two times. Return true if (time1 > time2) and false overwise.
*1h 15min is greater than 1h 10min.*

```
Time operator- (const Time&) const;
```
Overloaded operator – subtracts two times. The value of time cannot be negative. If subtrahend value greater than minuend the result should be equal to 0h 0min.
*1h 15min – 0h 20 min = 0h 55min; 1h 20min – 1h 30 min = 0h 0min.*

Test your functions with the given main function (testTime.cpp).

Here is a sample run:

```
3 hours, 55 minutes

3 hours, 45 minutes

3 hours, 20 minutes

0 hours, 10 minutes

0 hours, 0 minutes

7 hours, 15 minutes
```

3) Design a class named Rectangle with two private data fields named width and height. Implement the relational operators (<, =>, ==, !=) in the Rectangle class to compare Rectangle objects (2 objects) according to their area. The functions for relational operators should return Boolean data type. Write all code in one source file, no need to split code for design and implementation. Upload it to as a .cpp file under homework-5 in the Canvas. Use Your First Name + Your Last Name + Part2 + HW4. For example, Bayazit_Karaman_Part3_HW4.cpp.