**Lab test 3**

**Full marks – 40**

**Today’s Topics**

* Class
* Object
* Inheritance
* encapsulation
* method override
* abstract class

**Seasons of Bangladesh**

Create a system where we will work with the temperature of seasons. Implement the system in such a way where users can increase the temperature of a season, decrease the temperature of a season, display the temperature of a season and display the info of a season. Here we will work with 2 seasons

1. **Summer**
2. **Winter**

Both of the seasons will have three common attributes.

1. Season name (this will be a string)
2. Months ( this will be a list)
3. Year

Note #1:

1. Maximum temperature for summer is 40 degree Celsius.
2. Minimum temperature for summer is 25 degree Celsius.

Note #2:

1. Maximum temperature for winter is 25 degree Celsius.
2. Minimum temperature for winter is 4 degree Celsius.

***What you need to do:***

1. Create an **abstract** class **Season** which will have **these private instance variables*; season\_name, months, year***
   1. Create a **constructor** that takes initial value for those attributes and initializes those attributes.

Create the **following methods as** described

1. ***display()***

* This method displays the attributes in this format-

Season name: season\_name

Months: months

Year: year

1. Create getter/setter methods for all these private attributes.
2. Create an **abstract** method ***increaseTemperature()***
3. *Create another* ***abstract*** *method* ***decreaseTemperature()***
4. *Create another* ***abstract*** *method* ***getTemperature()***
5. Create a **Summer** class and make it a child of **Season** class. Now do the following.
   1. Create a **constructor** that takes initial value for attributes – ***season\_name, months, year, temperature***
   2. Inside the constructor, call the parent class’s constructor to pass ***season\_name, months, year***. Add a new private attribute ***temperature*** and set the ***temperature***. This is the initial temperature of the season.

Now you can increase and decrease the temperature by implementing the following two methods.

* 1. ***increaseTemperature( amount )***
* Check if the temperature crosses the maximum limit of 40 degree if increased.
* If it does not cross the limit, then increase the temperature by ***amount.***
* Otherwise print a statement “Maximum temperature limit reached.”
  1. ***decreaseTemperature( amount )***
* Check if the temperature crosses the minimum limit of 25 degree if decreased.
* If it does not cross the limit, then decrease the temperature by ***amount.***
* Otherwise print a statement “Minimum temperature limit reached.”
  1. ***getTemperature()***
* return the temperature
  1. ***display()***
* display the followings –

Season name: season\_name

Months: months

Year: year

Temperature: temperature

1. Create a **Winter** class and make it a child of **Season** class. Now do the following.
   1. Create a **constructor** that takes initial value for attributes – ***season\_name, months, year, temperature***
   2. Inside the constructor, call the parent class’s constructor to pass ***season\_name, months, year***. Add a new private attribute ***temperature*** and set the ***temperature***. This is the initial temperature of the season.

Now you can increase and decrease the temperature by implementing the following two methods.

* 1. ***increaseTemperature( amount )***
* Check if the temperature crosses the maximum limit of 25 degree if increased.
* If it does not cross the limit, then increase the temperature by ***amount.***
* Otherwise print a statement “Maximum temperature limit reached.”
  1. ***decreaseTemperature( amount )***
* Check if the temperature crosses the minimum limit of 4 degree if decreased.
* If it does not cross the limit, then decrease the temperature by ***amount.***
* Otherwise print a statement “Minimum temperature limit reached.”
  1. ***getTemperature()***
* return the temperature
  1. ***display()***
* display the followings –

Season name: season\_name

Months: months

Year: year

Temperature: temperature

1. Now in main method, for each of the classes do the followings-
2. create an object
3. print the season name
4. print the months
5. print the year
6. set the year
7. print the year
8. increase the temperature ( so that does not cross the limit)
9. print the temperature
10. decrease the temperature (so that does not cross the limit)
11. print the temperature
12. increase the temperature ( so that crosses the limit)
13. print the temperature
14. decrease the temperature (so that crosses the limit)
15. print the temperature
16. display all info

Rubrics –

* + - 1. Class and object – Writing the main method - 10
      2. Inheritance – Inheriting the abstract class and using its implemented methods - 10
      3. Method override – Inheriting the abstract class and overriding the necessary methods - 10
      4. Abstract class – Writing the abstract class and methods - 5

– Implementing the abstract methods - 5