**ASSIGNMENT SET 7**

**Date: Thursday, 10/04/2020**

**Submission Deadline: Sunday 17/04/2020**

**Objective(s): Learn to implement the following features…..**

* **Simple Inheritance,**
* **Multiple Inheritance**

**================================================================**

**Assignment 1 (requires concept of inheritance):**

**===================================================================**

class Tool

{ /\* Fill in \*/

};

/\*Implement class Scissors \*/

/\*Implement class Paper \*/

/\*Implement class Rock \*/

int main() {

// Example main function

// You may add your own testing code if you like

Scissors s1(5);

Paper p1(7);

Rock r1(15);

cout << s1.fight(p1) << p1.fight(s1) << endl;

cout << p1.fight(r1) << r1.fight(p1) << endl;

cout << r1.fight(s1) << s1.fight(r1) << endl;

return 0;

}

 In the above sample code, implement a class called Tool. It should have an int field called strength and a char field called type. You may make them either private or protected. The Tool class should also contain the function void setStrength(int), which sets the strength for the Tool.

Create 3 more classes called Rock, Paper, and Scissors, which inherit from Tool. Each of these classes will need a constructor which will take in an int that is used to initialize the strength field. The constructor should also initialize the type field using 'r' for Rock, 'p' for Paper, and 's' for Scissors.

These classes will also need a public function bool fight(Tool) that compares their strengths in the following way:

* Rock's strength is doubled (temporarily) when fighting scissors, but halved (temporarily) when fighting paper.
* In the same way, paper has the advantage against rock, and scissors against paper.
* The strength field shouldn't change in the function, which returns true if the original class wins in strength and false otherwise.

You may also include any extra auxiliary functions and/or fields in any of these classes. Run the program without changing the main function, and verify that the results are correct.

**Assignment 2 (Learn to implement multiple inheritance)**

**===================================================================**

Implement the Employee-(AdminOfficer,Faculty)-Dean polymorphic hierarchy as discussed in the last Lecture with suitable CalcSalary function (consider some suitable components). Make Employee class as Abastract base class.