(ASSIGNMENT QUESTIONS)

(MADE BY ALI AKBER)

(BSCS 2ND SS1)

// COMPARING TWO RECTANGLES.

```
#include <iostream>
#include <string.h>
using namespace std;
class Rectangle{
        private:
        int length,width;
        string color;
public:
Rectangle():length(0),width(0),color("Null"){
}
Rectangle(int l,int w,    string c) {
        length=l;
        width=w;
        color=c;
}
void getdata(){
        cout<<"Enter length of rectangle:"<<endl;
        cin>>length;
        cout<<"Enter width of rectangle: "<<endl;
        cin>>width;
        cout<<"Enter color of rectangle:"<<endl;
        cin>>color;
```

```
}
int getarea(){
        return length*width;
}
string getcolor(){
        return color;
}

void showdata(){
        cout<<"Length of rectangle is:"<<length<<endl;
        cout<<"Width of rectangle:"<<width<<endl;
        cout<<"Color of rectangle is:"<<color<<endl;
        cout<<"area of rectangle is:"<<getarea()<<endl;
}
};


int main (){
Rectangle r1,r2,r3;
cout<<"Enter details of rectangle 1:"<<endl;
r1.getdata();
cout<<"Enter details of rectangle 1:"<<endl;
r2.getdata();
r1.getarea();
r2.getarea();
```

```
if (r1.getarea()==r2.getarea() && r1.getcolor()==r2.getcolor()){
        cout<<"Given rectangles are matched:"<<endl;
}
else{
            cout<<"Given rectangles are not matched:"<<endl;
}
return 0;
}
```

//ADDITION OF TWO COMPLEX NUMBER

//WITHOUT OPERATOR OVERLOADING.

```
#include <iostream>
using namespace std;
class Complex{
        private:
                int real;
                int imaginary;
                public:
                        Complex (){
                                real=0;
                                imaginary=0;
                        }
                        Complex (int r, int img){
                                real=r;
                                imaginary=img;
                        }
                        void getComplex(){
```

```
                cout<<"Enter real part of complex number:"<<endl;
                cin>>real;
                        cout<<"Enter imaginary part of complex number:"<<endl;
                cin>>imaginary;
        }
        void showComplex(){
                cout<<" Real number is:"<<real<<endl;
                cout<<" Imaginary number is:"<<imaginary<<endl;
        }
          Complex AddComplex(Complex c3){
          Complex temp;
temp.real=real+c3.real;
temp.imaginary=imaginary+c3.imaginary;
return temp;
          }
          void getresult(){
                cout<<"Addition of two complex number
gives:"<<real<<"+"<<imaginary<<"i"<<endl;
          }
};
int main (){
Complex c1,c4;
cout<<"Ist Complex number is given as:"<<endl;
c1.getComplex();
c1.showComplex();
cout<<endl;
```

```
Complex c2(12,7);
c2.showComplex();
cout<<endl;
c4=c1.AddComplex(c2);
c4.getresult();
        return 0;
}
```

//ADDITION OF TWO COMPLEX NUMBER

//WITH OPERATOR OVERLOADING.

```
#include <iostream>
using namespace std;
class Complex{
        private:
                int real;
                int imaginary;
                public:
                        Complex (){
                                real=0;
                                imaginary=0;
                        }
                        Complex (int r, int img){
                                real=r;
                                imaginary=img;
                        }
                        void getComplex(){
                                cout<<"Enter real part of complex number:"<<endl;
```

```
                cin>>real;
                        cout<<"Enter imaginary part of complex number:"<<endl;
                cin>>imaginary;
        }
        void showComplex(){
                cout<<" Real number is:"<<real<<endl;
                cout<<" Imaginary number is:"<<imaginary<<endl;
        }
          Complex operator +(Complex c3){
                  Complex temp;
temp.real=real+c3.real;
temp.imaginary=imaginary+c3.imaginary;
return temp;
          }
          void getresult(){
                cout<<"Addition of two complex number
gives:"<<real<<"+"<<imaginary<<"i"<<endl;
          }
};
int main (){
Complex c1,c4;
cout<<"Ist Complex number is given as:"<<endl;
c1.getComplex();
c1.showComplex();
cout<<endl;
Complex c2(12,7);
```

```
c2.showComplex();
cout<<endl;
c4=c1+c2;
c4.getresult();
        return 0;
}
```

//ADDITION OF TWO RATIONAL NUMBER

//WITHOUT OPERATOR OVERLOADING.

```
#include <iostream>
using namespace std;
class Rational{
        private:
                int num;
                int dnum;
                public:
                        Rational (){
                                num=0;
                                dnum=0;
                        }
                        Rational(int a, int b){
                                num=a;
                                dnum=b;
                        }
                        void getRational(){
                                cout<<"Enter numerator of rational number:"<<endl;
                                cin>>num;
```

```
                                        cout<<"Enter denominator of rational number:"<<endl;
                                cin>>dnum;
                        }
                        void showRational(){
                                cout<<" Numerator number is:"<<num<<endl;
                                cout<<" Denominator number is:"<<dnum<<endl;
                        }
                          Rational AddRational(Rational r3){
                                  Rational temp;
temp.num=num*r3.dnum+r3.num*dnum;
temp.dnum=dnum*r3.dnum;
return temp;
                          }
                          void getresult(){
                                cout<<"Addition of two rational number
gives:"<<num<<"/"<<dnum<<endl;
                          }
};
int main (){
Rational r1,r4;
cout<<"Ist Rational number is given as:"<<endl;
r1.getRational();
r1.showRational();
cout<<endl;
Rational r2(12,7);
r2.showRational();
```

```
cout<<endl;
r4=r1.AddRational(r2);
r4.getresult();
        return 0;
}
```

//ADDITION OF TWO RATIONAL NUMBER

//WITH OPERATOR OVERLOADING.

//ASSIGNMENT QUESTION.

```
#include <iostream>
using namespace std;
class Rational{
        private:
                int num;
                int dnum;
                public:
                        Rational (){
                                num=0;
                                dnum=0;
                        }
                        Rational(int a, int b){
                                num=a;
                                dnum=b;
                        }
                        void getRational(){
                                cout<<"Enter numerator of rational number:"<<endl;
                                cin>>num;
```

```
                cout<<"Enter denominator of rational number:"<<endl;
        cin>>dnum;
    }
    void showRational(){
        cout<<" Numerator number is:"<<num<<endl;
        cout<<" Denominator number is:"<<dnum<<endl;
    }
     Rational operator+(Rational r3){
         Rational temp;
temp.num=num*r3.dnum+r3.num*dnum;
temp.dnum=dnum*r3.dnum;
return temp;
     }
      Rational operator*(Rational r3){
         Rational temp;
temp.num=num*r3.num;
temp.dnum=dnum*r3.dnum;
return temp;
     }
     void RationalAddition(){
        cout<<"Addition of two rational number
gives:"<<num<<"/"<<dnum<<endl;
     }
     void RationalMultiplication(){
        cout<<"Addition of two rational number
gives:"<<num<<"/"<<dnum<<endl;
     }
```

```
};
int main (){
Rational r1,r4,r5;
cout<<"Ist Rational number is given as:"<<endl;
r1.getRational();
r1.showRational();
cout<<endl;
Rational r2(4,5);
r2.showRational();
cout<<endl;
r4=r1+r2;
r4.RationalAddition();
r5=r1*r2;
r5.RationalMultiplication();
        return 0;
}
```

//DERIVED CLASS CONSTRUCTOR.

//ASSIGNMENT QUESTION.

```
#include <iostream>
#include<string.h>
using namespace std;
class Person{
        private:
                int id;
                char name[50];
                public:
```

```
                Person(){
                id=0;
                strcpy(name,"");        }       //no argument constructor.
                Person(int i,char na[]){
                        id=i;
                        strcpy(name,na);
                }                                               //two arg constructor.
                void showdata1(){
                        cout<<"Id is :"<<id<<endl;
                                cout<<"Name is :"<<name<<endl;
                }
                void getdata1(){
                        cout<<"Enter id of the person:"<<endl;
                        cin>>id;
                        cout<<"Enter name of the person:"<<endl;
                        cin>>name;
                }
};
class Teacher:public Person{
        private:
                float salary;
                char publication[50];
                public:
                Teacher():Person(){
                salary=0;
                strcpy(publication,"");    }
```

```
        Teacher(int i,char na[],float sal,char pub[])
            :Person(i,na){
            salary=sal;
            strcpy(publication,pub);    };
            void getdata2(){
                    cout<<"Enter salary of the teacher:"<<endl;
                    cin>>salary;
                    cout<<"Enter publication of the teacher:"<<endl;
                    cin>>publication;
            }
            void showdata2(){
                    cout<<"Salary of the teacher is:"<<salary<<endl;
                    cout<<"Publication of the teacher is:"<<publication<<endl;
            }
};
int main (){
Teacher t1;
t1.getdata1();
t1.getdata2();

cout<<endl;
t1.showdata1();
t1.showdata2();
        return 0;
}
```