**Warm-Up Part a**

**Main:**

//int main(int argc, char \*argv[])

int main(){

Rational r1 = Rational(1, 2);

Rational r2 = Rational(500, 1000);

//Rational r = r1.mult(r2);

Rational r = r1 \* r2;

cout<<r.numer()<<"/"<<r.denom()<<endl;

if(r1==r2){

cout<<"They match"<<endl;

}

else{

cout<<"They don't match"<<endl;

}

return 0;

}

**Overloaded == operator:**

bool operator == (Rational lhs, Rational rhs){

int lNum = lhs.numer();

int rDenom = rhs.denom();

int rNum = rhs.numer();

int lDenom = lhs.denom();

int divideBy;

int newNum;

int newDenom;

if(lDenom == rDenom && lNum == rNum){

return true;

}

else if (lDenom > rDenom){

if (lDenom%rDenom == 0){

divideBy = lDenom/rDenom;

newNum = lNum/divideBy;

newDenom = lDenom/divideBy;

if(newDenom == rDenom && newNum == rNum){

return true;

}

}

}

else if (rDenom > lDenom){

if (rDenom%lDenom == 0){

divideBy = rDenom/lDenom;

newNum = rNum/divideBy;

newDenom = rDenom/divideBy;

if(newDenom == lDenom && newNum == lNum){

return true;

}

}

}

return false;

}

**Result:**

500/2000

They match

**Part B:**

**Main:**

int main(){

Rational r1 = Rational(1, 2);

Rational r2 = Rational(3, 4);

//Rational r = r1.mult(r2);

Rational r = r1 \* r2;

cout<<r.numer()<<"/"<<r.denom()<<endl;

/\* if(r1==r2){

cout<<"They match"<<endl;

}

else{

cout<<"They don't match"<<endl;

}\*/

r2+=r1;

cout<<r2.numer()<<endl;

cout<<r2.denom()<<endl;

return 0;}

**Overloaded += Operator:**

void Rational::operator +=(Rational rhs){

int tmpDenom = rhs.denom()\*denom\_;

int tmpNumerator = (rhs.numer()\*denom\_)+(rhs.denom()\*numer\_);

numer\_ = tmpNumerator;

denom\_ = tmpDenom;

}

**Result:**

FHosts-MacBook-Pro:Rational fhost$ g++ \*.cpp -o main

FHosts-MacBook-Pro:Rational fhost$ ./main

3/8

10

8

**Part c:**

**Main:**

int main(){

Rational r1 = Rational(1, 2);

Rational r2 = Rational(500, 1000);

//Rational r = r1.mult(r2);

Rational r = r1 \* r2;

cout<<r.numer()<<"/"<<r.denom()<<endl;

/\* if(r1==r2){

cout<<"They match"<<endl;

}

else{

cout<<"They don't match"<<endl;

}

r2+=r1;

cout<<r2.numer()<<endl;

cout<<r2.denom()<<endl;

\*/

r2.reduce();

cout<<r2.numer()<<endl;

cout<<r2.denom()<<endl;

return 0;

}

**Reduce:**

void Rational::reduce(){

int tmpNumer = numer\_;

int tmpDenom = denom\_;

int gCd;

//computing GCD

while (tmpNumer != 0 && tmpDenom != 0){

if(tmpNumer > tmpDenom){

tmpNumer = tmpNumer%tmpDenom;

}

else{

tmpDenom = tmpDenom%tmpNumer;

}

}

if(tmpDenom > tmpNumer){

numer\_ = numer\_/tmpDenom;

denom\_ = denom\_/tmpDenom;

}

else{

denom\_ = denom\_/tmpNumer;

numer\_ = numer\_/tmpNumer;

}

//numer\_ = tmpNumer;

// denom\_ = tmpDenom;

}

**Result:**

500/2000

1

2

FHosts-MacBook-Pro:Rational fhost$