جَامعة
المَنارة
MANARA UNIVERSITY

مقرر برمجة ٢
الجلسة السابعة عملي

النسخ الافتراضي للأغراض (Default copying for objects):

أوجد خرج البرنامج التالي:

```
#include <iostream>
using namespace std;
class CRectangle {
int width, height;
public:
CRectangle (int a,int b);
int area () {return (width*height);} };
CRectangle::CRectangle (int a, int b) {
width = a;
height = b;}
int main() {
CRectangle rect (3,4);
cout << "rect area: " << rect.area() << endl;
CRectangle rect1=rect;
cout << "rect1 area: " << rect1.area() << endl;
CRectangle rect2(rect1);
cout << "rect2 area: " << rect2.area() << endl;
return 0;}
```

```
rect area:12
rect area:12
rect area:12
```

التحميل الزائد للمعاملات باستخدام التوابع الأعضاء:

```
#include <iostream>
#include <iomanip>
using namespace std;
class Time {
public: Time(int ,int ,int ,char) ; //constructor
void printUniversal(); //printing Time
private:
int hour; // 0 - 23 (24-hour clock format)
int minute; // 0 - 59
int second; // 0 - 59
char sep; //:
};
Time::Time(int hr,int mn,int sc,char sp) {
hour=(hr>23)?hr/24:hr; minute=(mn>59)?mn/60:mn;
second=(sc>59)?sc/60:sc; sep=(sp==':')?sp:':';
}
void Time::printUniversal() { cout << setfill( '0') <<
setw( 2) << hour << sep<<< minute << sep << setw( 2)
<< second<<endl;}
```

حمل المعامل + بشكل زائد ليجمع غرضين من نوع Time .
حمل المعامل - بشكل زائد ليطرح غرضين من نوع Time .
حمل المعامل > بشكل زائد ليجمع غرضين من نوع Time .
وذلك باعتبار أن الصف Time معرف بالشكل المرفق.

التحميل الزائد للمعاملات باستخدام التوابع الأعضاء:

```
Time Time::operator +(Time T)
{ Int t=0;
 t=second+T.second;          second=t>59?t%60:t;
t=(t/60)+minute+T.minute;   minute=t>59?t%60:t;
t=(t/60)+hour+T.hour;          hour=t>23?t%24:t;
 return *this;}
Time Time::operator - (Time T)
{ if (second>=T.second)
          second=second-T.second;
else  { second=(second+60)-T.second; minute--; }
if (minute>=T.minute)
         minute=minute-T.minute;
else
          {minute=(minute+60)-T.minute; hour--; }
if(hour >=T.hour)
            hour=hour - T.hour;
else        hour=( hour+24)-T. hour;
         return *this;
 }
```

```
bool Time::operator >(Time T)
{ if (hour>T.hour)       return 1;
 else if (hour<T.hour)       return 0;
if (minute>T.minute)     return 1;
 else if (minute<T.minute) return 0;
if(second>=T.second)       return 1;
Else      return 0; }
```

التحميل الزائد للمعاملات باستخدام التوابع الأعضاء:

```
int main()
{ Time t1(10,20,30,':');
Time t2(15,25,35,':');
 Time t3(0,0,0,':');
cout<<"t1 in the initial state = ";       t1.printUniversal();
cout<<"t2 in the initial state = ";        t2.printUniversal();
cout<<"t3 in the initial state = ";        t3.printUniversal();
t3=t1+t2; //calling the overloaded +
cout<<"\n\nt1 after calling the + operator = ";  t1.printUniversal();
cout<<"t2 after calling the + operator = ";        t2.printUniversal();
cout<<"t3 after calling the + operator = ";        t3.printUniversal();
if (t1>t2)   t3=t1-t2; //calling the overloaded –
else          t3=t2-t1; //calling the overloaded –
   cout<<"\n\nt1 after calling the - operator = "; t1.printUniversal();
cout<<"t2 after calling the - operator = ";
 t2.printUniversal();
cout<<"t3 after calling the - operator = ";
 t3.printUniversal();
system ("pause");
return 0;
}
```

التحميل الزائد للمعاملات باستخدام التوابع الأعضاء:

أنشئ صف يمثل عدد عقدي Complex يملك البيانات الأعضاء الخاصة الجزء الحقيقي والجزء التخيلي.
أنشئ باني يقوم بتهيئة المتحولات الأعضاء.
أنشئ تابع عضو لطباعة العدد العقدي بالشكل real+image i.
أنشئ تابع عضو لتحميل المعامل + ليجمع عددين عقديين.

```
#include<iostream>
using namespace std;
class Complex {
private:    int real, imag;
public:
Complex(int r = 0, int i = 0) {real = r; imag = i;}
void print() { cout << real << " + i" << imag << endl; }
 Complex operator + (Complex c1)
{Complex res;
res.real = c1.real + real;
res.imag = c1.imag + imag;
return res;}
};
int main()
{Complex c1(10, 5), c2(2, 4);
Complex c3 = c1 + c2;
c3.print();
return 0;}
```

التحميل الزائد للمعاملات باستخدام التوابع الصديقة:

```
#include<iostream>
using namespace std;
class Complex {
private:    int real, imag;
public:
Complex(int r = 0, int i = 0) {real = r; imag = i;}
void print() { cout << real << " + i" << imag << endl; }
friend Complex operator + (Complex,Complex);};
Complex operator + (Complex c1,Complex c2)
{Complex res;
res.real = c1.real + c2.real;
res.imag = c1.imag + c2.imag;
return res;}
int main()
{Complex c1(10, 5), c2(2, 4);
Complex c3 = c1 + c2;
c3.print();return 0;}
```

عدل على الكود السابق ليصبح تابع تحميل المعامل + تابع صديق يجمع عددين عقديين.

جَامعة
المَنارة
MANARA UNIVERSITY

انتهت الجلسة