جَامعة
المَنارة
MANARA UNIVERSITY

مقرر برمجة ٢
الجلسة السادسة عملي

```
#include <iostream>
using namespace std;
class CRectangle
 {  int width, height;
  public:
void set_values (int, int);
 int area (void) {return (width * height);} };
 void CRectangle::set_values (int a, int b) {  width = a;  height = b; } int main() {
 CRectangle a, *b, *c;
 CRectangle * d = new CRectangle[2];
 b= new CRectangle;
 c= &a;  a.set_values (1,2);
b->set_values (3,4);
  d->set_values (5,6);
 d[1].set_values (7,8);
cout << "a area: " << a.area() << endl;
 cout << "*b area: " << b->area() << endl;
 cout << "*c area: " << c->area() << endl;
cout << "d[0] area: " << d[0].area() << endl;
cout << "d[1] area: " << d[1].area() << endl;
delete[] d;  delete b;
  return 0;}
```

التخصيص الديناميكي للذاكرة

أوجد خرج البرنامج التالي:

```
a area: 2
*b area: 12
*c area: 2
d[0] area: 30
d[1] area: 56
```

التحميل الزائد للتوابع:

أوجد خرج البرنامج التالي:

```
##include <iostream>
using namespace std;

class DD
{public:
   int DD_Function(int i){return i;}
   double DD_Function(double d){return d;}
};
int main()
{DD obj;
cout<<obj.DD_Function(100)<<endl;
cout<<obj.DD_Function(5005.56);
return 0;
}
```

```
100
5005.56
```

تمرين:

اكتب برنامج C++ لإنشاء صنف يسمى BankAccount والذي يحتوي على المتغيرات الأعضاء الخاصة :
رقم الحساب والرصيد.
إنشاء باني افتراضي للصنف وباني ثاني مع بارامترات.
تضمين وظائف member functions لإيداع وسحب الأموال من الحساب.
تابع لمقارنة حسابين و إعادة الحساب ذو الرصيد الأعلى.
كما يتضمن تابع صديق لإضافة ٣٠% على حساب معين إذا كان الحساب أقل من ١٠٠٠٠٠٠
في التابع الرئيسي أنشئ حساب بمبلغ معين واسحب منه ٥٠٠٠٠ .
أنشئ حساب آخر وقارن رصيدهما وأعد الحساب ذو الرصيد الأعلى.

تمرين:

```
#include <iostream>
#include <string>
using namespace std;
class BankAccount
{
private:
string accountNumber;
 int balance;
public:
  // Constructor and deconstructors
 BankAccount (string  accNum,int initialBalance):accountNumber (accNum),
    balance (initialBalance)
  {    }
BankAccount(){cout<<“new account is created”;}

~BankAccount(){cout<<“The account is deleted”;}
```

```
// Member function to deposit money
  void deposit (int amount)
  { balance += amount;
    cout << "Deposit successful. Current balance: " <<
balance << endl; }

// Member function to withdraw money
  void withdraw (int amount)
  { if (amount <= balance)
      {balance -= amount;
              cout << "Withdrawal successful. Current
balance: " << balance << endl; }
    else
      {cout << "Insufficient balance. Cannot withdraw." <<
endl  }
  }
```

```
friend void inc_balance (BankAccount &B);
BankAccount compare_acc (BankAccount B1, BankAccount B2)
  {  if (B1.balance > B2.balance)
      {balance=B1.balance; return B1;}
     else
      { balance=B1.balance; return B2;}  }
  void print(){
    cout<<"accountNumber: "<<accountNumber <<"\tbalance:
"<<balance <<endl;}
};
//implement friend function inc_balance
  void inc_balance (BankAccount &B)
  {if (B.balance<=2000000)
      B.balance += (0.3*B.balance); }
```

```
int main ()
{
   BankAccount BA("1234",2000000);
   BA.print();
   BA.withdraw(50000);
   BA.print();
   inc_balance(BA);
   BA.print();
   BankAccount BA1("1250",1500000);
   BankAccount BA2("11230",0);
   BA2.compare_acc(BA,BA1);
   cout<<"the max balance: ";
   BA2.print();
  return 0;
}
```

يمكن تحقيق الطريقة compare_acc بطريقة أخرى:

```
BankAccount compare_acc (BankAccount B1)
 {  if (balance > B1.balance)
    return *this;
   else
return B1;}
```

ويكون الاستدعاء بالشكل:

```
BankAccount BA1("1250",1500000);
   BankAccount BA2("11230",0);
 BankAccount BA=BA1. compare_acc(BA2);
   cout<<"the max balance: ";
   BA.print();
```

تمرين إضافي:

أنشئ صنف يمثل هاتف phon له الخصائص الموديل model وسنة الإصدار year.
أضف بواني وهادم للصنف.
أضف طريقة لإدخال بيانات الهاتف طريقة لطباعتها.
أضف طريقة لمعرفة الهاتف ذو الإصدار الأقدم

```
#include <iostream>
using namespace std;
class phone  {
private: int model; int  year;
public:
   phone();
   phone(int ,int );
   ~phone();
   void input (){cin>> model>> year ;}
   void output (){cout << model << endl<< year<< endl;}
  phon compare(phone p)
   { if(yrea<p.year)
            return *this;
      else return p;
    }
 };
```

تمرين إضافي:

```
phone :: phone (){model=year=1995;}
phone :: phone (int a , int b ){model=a;year=b;}
phone ::~phone()
{cout << "the object is deconstructed"<<year<<endl;}

int main() {
            phone huawei (44,2004) ;
            huawei.output ();
            phone samsung ;
           samsung.output();
           phone p= samsung. compare(huawei);
           p. output ();
        return 0;
}
```

جَامعة
المَنارة
MANARA UNIVERSITY

انتهت الجلسة