جامعة القدس المفتوحة

كلية لتكنولوجيا والعلوم التطبيقية

الدليل العملي لمقرر

برمجة (1) رقم 1291

إعداد

د. يوسف ابو زر أ. عيسى أسعد

2013

المحتويات

1.االمختبر العملي الاول: التراكيب والأصناف

- التراكيب : الشخص Person
- التراكيب : النقطة Point
- الصنف : النقطة Point
- البناء Constructor

2- المختبر الثاني المصفوفات والاصناف Arrays and Classes

- المؤشرات و مصفوفة من الكائنات
- القوائم المتصلة

3- المختبر الثالث الدوال والأصناف الصديقةFriend Functions and Operators

- الطرق المختلفة لتمرير العوامل
- الدوال الصديقة Friend Functions
- الأصناف الصديقة

4- المختبر الرابع الأصناف المشتقة والقوالبDerived Classes and Templates

- الوراثة المنفردة Single Inheritance
- الوراثة المتعددة Multiple Inheritance
- تعدد الاوجه Polymorphism
- القوالب Templates

5- المختبر الخامس معالجة الملفات بلغة ++C File Processing with C++

6-الملاحق مترجمات لغة++C

- إعداد بيئة تطوير ضمن برنامج Code::Blocks
- مقدمة
- إعداد بيئة تطوير ضمن برنامج Code::Blocks.
- تحميل برنامج Code::Blocks.
- تنصيب برنامج Code::Blocks على الحاسوب.
- شرح واجهة التطبيق
- تشغيل برنامج Code::Blocks من خلال إنشاء مشروع جديد و تطبيق مثال بسيط

- إعداد بيئة تطوير ضمن برنامج Visual C++.

الوحدة الثانية
التراكيب والأصناف
Structures and Classes

في هذا المختبر ، عزيزي الطالب، سيعرض عضو هيئة التدريس في فرع الجامعة بعض التدريبات العملية لمفهومين مهمين وهما التراكيب والأصناف. تقابل التراكيب Structures السجلات Records في لغات برمجة أخرى. أما الأصناف Classes فهي الفكرة الرئيسة في لغات البرمجة الكينونية Object Oriented Languages وهي جزء من لغة C++ وليست جزءا من لغة C الأصلية.

كما سيعرض برامج بلغة C++ تناقش مفاهيم رئيسة أخرى في لغات البرمجة الكينونية، مثل الكائنات والمتغيرات والدوال المنتمية وكيفية تعريفها واستخدامها في لغة C++. وتعتبر هذه المفاهيم أساسية لفهم بقية التدريبات في هذا المقرر، لذا نرجو منك، عزيزي الطالب، التأكد من تنفيذ وفهم هذه البرمجيات في المختبر بإشراف عضو هيئة التدريس قبل الانتقال إلى بقية الوحدات. وقد تم تزويدك في تدريبات هذه الوحدة بالعديد من الأمثلة والتمارين التي نرجو منك إعطاءها الوقت والجهد اللازمين لحلها وتنفيذها. تذكر، عزيزي الطالب، أن أجوبة هذه التمارين موجودة في نهاية هذه الوحدة، ولكن وحتى تتحقق الفائدة القصوى عليك، عزيزي الطالب ، أن تحاول بكل جد حل هذه التمارين وتنفيذها على الحاسوب قبل النظر إلى الحل.

أهداف التدريبات

ينتظر منك، عزيزي الطالب، بعد فراغك من تنفيذ هذه التدريبات أن تكون قادرا على أن:

1. تعريف واستخدام التراكيب structures.
2. استخدام مصفوفات من التراكيب arrays of structures.
3. تعريف واستخدام الأصناف classes.
4. التمييز بين التراكيب والأصناف.
5. إنشاء الكائنات واستخدامها بطريقة صحيحة.
6. كتابة الدوال المنتمية واستدعائها بطريقة صحيحة.
7. كتابة الدوال السطرية .
8. تحميل الدوال أكثر من تعريف واحد.
9. إدخال البيانات وإخراجها بوساطة cin و cout.

التدريب رقم (1)

عرف تركيب يصف الشخص Person يتكون من الحقول التالية: name الاسم ، age العمر ، gender الجنس، واكتب البرنامج الرئيسي الذي يعرف متغير من النوع شخص ويقوم بإسناد قيم للمتغيرات وطباعة النتائج.

الحل:

```
#include <iostream>
#include <string>
struct Person{
	string name;
	int age;
	char gender;
};
int main(){
	Person p;
	p.name = "Christopher";
	p.age = 34;
	p.gender = 'M';
	cout << "Name: " << p.name << endl;
	cout << "Age: " << p.age << endl;
	cout << "Gender: " << p.gender << endl;
	return 0;
}
```

التدريب رقم (2)

عرف تركيب يصف النقطة Point تتكون إحداثيات x و y، واكتب البرنامج الرئيسي الذي يعرف متغير مؤشري من النوع نقطة ويقوم بإسناد قيم للمتغيرات وطباعة النتائج.

الحل:

```
include <iostream>
struct Point{
        int x;
        int y;
};
int main(){
        Point* p = new Point;
        p->x = 9;
        p->y = 4;
        cout << p->x << " " << p->y << endl;
        return 0;
}
```

اسئلة للحل على الحاسوب (1)

عزيزي الطالب اعد كتابة التدريب الســـابق بحيث نعرف مصـــفوفة تتكون من 6 نقاط وإيجاد المسافة بين كل نقطتين حسب العلاقة التالية:

المسافة = $(x^2 + y^2)^{0.5}$

مساعدة الحل:

1-تعريف المصفوفة

```
.Point pts[6];  // array to hold up to 6 points
```

2-الاقتران لإيجاد المسافة بين النقاط:

```
float dist(Point p) {
   // Compute the distance from the origin
   return sqrt(p.x*p.x + p.y*p.y);
```

جملة الاستدعاء dist(pt[i])

التدريب رقم (3)

اعد كتابة التدريب السابق النقطة Point تتكون إحداثيات x و y باستخدام الأصناف class ، واكتب البرنامج الرئيسي الذي يعرف متغير مؤشري من النوع نقطة ويقوم بإسناد قيم للمتغيرات وطباعة النتائج.

الحل:

```
#include <iostream>
class Point{
public:
	int x, y;
	Point(){
		x = 0; y = 0;
	}
};
int main(){
	Point* p = new Point;
	p->x = 9;
	p->y = 4;
	cout << p->x << " " << p->y << endl;
	return 0;
}
```

التدريب رقم (4)

ادرس المثال التالي جيدا والذي يصف صنف من نوع مستطيل Rectangle ويقوم بإعطاء إبعاد المستطيل وإيجاد مساحته باستخدام دالة سطريه وطباعة النتائج. اعد كتابة المثال السابق باستخدام البناء constructor

```
#include <iostream>
class CRectangle {
   int x, y;
  public:
   void set_values (int,int);
   int area () {return (x*y);}
};

void CRectangle::set_values (int a, int b) {
  x = a;
  y = b;
}

int main () {
  CRectangle rect, rectb;
  rect.set_values (3,4);
  rectb.set_values (5,6);
```

```
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

الحل:

```
// باستخدام constructor
#include <iostream>
class CRectangle {
    int width, height;
  public:
    CRectangle (int,int);
    int area () {return (width*height);}
};

CRectangle::CRectangle (int a, int b) {
  width = a;
  height = b;
}

int main () {
  CRectangle rect (3,4);
  CRectangle rectb (5,6);
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

اسئلة للحل على الحاسوب (2)

عزيزي الطالب اعد كتابة التدريب السابق بحيث نعرف الهدام

مساعدة الحل:

الهدام : ;() ~CRectangle

اسئلة للحل على الحاسوب (3)

عزيزي الطالب، اكتب برنامج بلغة C++ لتعريف صـــنف طالب STUDENT يتكون من المتغيرات المنتمية التالية : الرقم الجامي للطالب USN ، الاسم name، وعلامات m في 3 اختبارات في موضوع ما, .والمعدل avg . عرف مصفوفة تمثل مجموعة من 10 كائنات من الصنف.STUDENT . باسـتخدام اقترانات منتمية مناسـبة، اوجد متوسـط اعلى علامتين علامات لكل طالب . وأطبع النتائج مثل: USN، والاسم ومعدل علامات الطلاب كافة.

التدريب رقم (5)

ادرس التدريب رقم (4) واعد كتابة البناء من خلال مفهوم اعادة التحميل.

الحل:

```
// overloading class constructors
#include <iostream>

class CRectangle {
    int width, height;
  public:
    CRectangle ();
    CRectangle (int,int);
    int area (void) {return (width*height);}
};

CRectangle::CRectangle () {
  width = 5;
  height = 5;
}

CRectangle::CRectangle (int a, int b) {
  width = a;
  height = b;
}

int main () {
  CRectangle rect (3,4);
  CRectangle rectb;
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

إجابات الأسئلة

سؤال (1)

```
struct Point {
   float x;   // x coordinate
   float y;   // y coordinate
};

Point pts[6];  // array to hold up to 6 points.
int   n = 0;      // number of points in the array.

while (cin >> pts[n].x >> pts[n].y) {
   n++;
}

float dist(Point p) {
   // Compute the distance from the origin
   return sqrt(p.x*p.x + p.y*p.y);
}

void findDistance(Point pt[], int size) {
   for (int i=0; i<size-1; i++) {
      cout<< dist(pt[i
   }
}
```

سؤال (2)

```
// Solution on constructors and destructors
#include <iostream>
class CRectangle {
   int *width, *height;
  public:
   CRectangle (int,int);
   ~CRectangle ();
   int area () {return (*width * *height);}
};

CRectangle::CRectangle (int a, int b) {
  width = new int;
  height = new int;
```

```
  *width = a;
  *height = b;
}

CRectangle::~CRectangle () {
  delete width;
  delete height;
}

int main () {
  CRectangle rect (3,4), rectb (5,6);
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

نتيجة التنفيذ

rect area: 12
rectb area: 30

سؤال (3)

```
#include<iostream.h>
#include<conio.h>
class student
{
        private :
                int usn,m[4];
                char name[20];
                float avg;
        public :
                void read();
                void avgm();
                void print();
};
void student :: read()
{
        cout<<"Enter the USA no.  :";
        cin>>usn;
        cout<<"Enter the name of the student :";
        cin>>name;
        cout<<name<<endl;
        cout<<"Enter the marks of 3 subjects :";
        cin>>m[1]>>m[2]>>m[3];
        avgm();
}

void student :: avgm()
{
        if(m[1]<=m[2] && m[1]<=m[3])
                avg=(float)(m[2]+m[3])/2;
        if(m[2]<=m[1] && m[2]<=m[3])
                avg=(float)(m[1]+m[3])/2;
        if(m[3]<=m[1] && m[3]<=m[2])
                avg=(float)(m[1]+m[2])/2;
}
void student :: print()
{

     cout<<"\n"<<usn<<"\t"<<name<<"\t\t"<<avg;


     /*  cout<<"USA No.:"<<usn<<"\n";
         cout<<"Name   :"<<name<<endl;
         cout<<"Average marks :"<<avg<<"\n";  */
}
```

```
void main()
{
	int n,i;
	student s[10];
	clrscr();
	cout<<"Enter the no of students :";
	cin>>n;
	for(i=0;i<n;i++)
		s[i].read();
	cout<<"USN\tNAME\t\tAVVEAGE MARKS\n";
	for(i=0;i<n;i++)
		s[i].print();
getch();
}
```

الوحدة الثالثة

المصفوفات والأصناف
Arrays and Classes

إن معظم التطبيقات العملية تتطلب استخدام قوائم من الكائنات list of objects، وغالبا ما تنفذ هذه القوائم باستخدام مصفوفات من الكائنات Arrays of objects أو باستخدام قوائم متصلة من الكائنات linked lists of objects. في هذه الوحدة، عزيزي الطالب، سنناقش كيفية إنشاء واستخدام مصفوفات وقوائم متصلة من الكائنات من خلال أمثلة عملية وتدريبات مختلفة.

أهداف الوحدة

ينتظر منك، عزيزي الطالب، بعد فراغك من تنفيذ البرامج في هذه الوحدة، أن تكون قادرا على أن:

1. تعرف مصفوفات من الكائنات بلغة C++.
2. تمرر المصفوفات كعوامل إلى دوال مختلفة.
3. تستخدم هذه المصفوفات في تطبيقات مختلفة مثل عمليات الفرز والبحث.
4. تنشئ الكائنات بطريقة ديناميكية.
5. تتعامل مع الكائنات من خلال عناوينها ومتغيرات مؤشرة pointer variables.

6. أن تنشئ قوائم متصلة من الكائنات وتستخدمها في تطبيقات مختلفة.

التدريب (1)

اعد كتابة مثال المستطيل من خلال فكرة مؤشر الى الصنف ومؤشر الى مصفوفة من الكائنات.

الحل:

```
// pointer to classes example
#include <iostream>

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;
}
```

التدريب (2)

ادرس المثال التالي جيدا ونفذه على الحاسوب اجب عن الأسئلة التالية بالتعاون مع عضو هيئة التدريس في المختبر

```
#include <iostream.h>

class CAT
{
  public:
    CAT() { itsAge = 1; itsWeight=5; }
   ~CAT() {}                          // destructor
    int GetAge() const { return itsAge; }
    int GetWeight() const { return itsWeight; }
    void SetAge(int age) { itsAge = age; }

  private:
    int itsAge;
    int itsWeight;
};

int main()
{
  CAT * Family[500];
  int i;
  CAT * pCat;
  for (i = 0; i < 10; i++)
  {
    pCat = new CAT;
    pCat->SetAge(2*i +1);
    Family[i] = pCat;
  }

  for (i = 0; i < 10; i++)
  {
    cout << "Cat #" << i+1 << ": ";
    cout << Family[i]->GetAge() << endl;
  }
 return 0;
}
```

1. ما اسم الصنف _____
2. ما وظيفة الدالة التالية: CAT() { itsAge = 1; itsWeight=5; } _____
3. ما اسم البناء _____

4. هل يوجد هدام ، واذا كانت اجابتك بنعم فما هي الدالة الهدام. _____
5. ما هي المتغيرات المنتمية في الصنف _____
6. أعطي أمثلة على الدوال المنتمية _____
7. اكتب الجملة التي تعرف مصفوفة من المؤشرات للصنف CAT _____
8. ما الهدف من استخدام الجمل التالية: _____

```
pCat = new CAT;
pCat->SetAge(2*i +1);
```

9. ما هو pCat _____
10. ما وظيفة new _____
11. لماذا استخدمنا الرمز -> في الجملة الثانية _____
12. أعطي مثال على دالة سطرية _____
13. هل النتيجة التي حصلت عليها بعد التنفيذ هي:

```
Output: Cat #1: 1
Cat #2: 3
Cat #3: 5
```

التدريب (3)

ادرس المثال التالي والذي يوضح مثال النقطة بحيث يقدم فكرة استخدام خوارزميات الفرز bubbleSort لترتيب النقط تصاعديا حسب المسافة من نقطة الأساس. اكتب الدالة الرئيسية للبرنامج

```
include <iostream>
struct Point {
   float x;   // x coordinate
   float y;   // y coordinate
   char  name[20]; // name of the point
};

Point pts[10];  // array to hold up to 10 points.
int   n = 0;      // number of points in the array.

while (cin >> pts[n].name >> pts[n].x >> pts[n].y) {
   n++;
}

float dist(Point p) {
   // Compute the distance from the origin
   return sqrt(p.x*p.x + p.y*p.y);
}
```

```
void bubbleSort(Point pt[], int size) {
   bool doMore = true;
   while (doMore) {
      doMore = false;  // Assume no more passes unless exchange made.
      for (int i=0; i<size-1; i++) {
         if (dist(pt[i]) > dist(pt[i+1])) {
            // Exchange elements
            Point temp = pt[i]; pt[i] = pt[i+1]; pt[i+1] = temp;
            doMore = true;  // Exchange requires another pass.
         }
      }
   }
}
```

تدريب (4) *مثال على القوائم المتصلة*

اكتب برنامج بلغة C++ يعرف صنف من نوع CAT يحتوي على متغير منتمي itsAge والاقترانات المنتمية التي تدخل عمر القطة وترجع عمر ألقطة استخدم القوائم المتصلة لتمثيل المشكلة عن طريق تعريف عقدة من نوع Node تحتوي على الاقترانات اللازمة لإضافة معلومات القطة CAT الى القائمة المتصلة .

```
// Linked list simple implementation

    #include <iostream.h>

     // object to add to list
     class CAT
     {
     public:
       CAT() { itsAge = 1;}
       CAT(int age):itsAge(age){}
       ~CAT(){};
       int GetAge() const { return itsAge; }
    private:
       int itsAge;
    };

    // manages list, orders by cat's age!
    class Node
    {
    public:
       Node (CAT*);
       ~Node();
       void SetNext(Node * node) { itsNext = node; }
       Node * GetNext() const { return itsNext; }
       CAT * GetCat() const { return itsCat; }
```

```
  void Insert(Node *);
  void Display();
private:
  CAT *itsCat;
  Node * itsNext;
};


Node::Node(CAT* pCat):
itsCat(pCat),
itsNext(0)
{}

Node::~Node()
{
  cout << "Deleting node...\n";
  delete itsCat;
  itsCat = 0;
  delete itsNext;
  itsNext = 0;
}

// **********************************
// Insert
// Orders cats based on their ages
// Algorithim: If you are last in line, add the cat
// Otherwise, if the new cat is older than you
// and also younger than next in line, insert it after
// this one. Otherwise call insert on the next in line
// **********************************
void Node::Insert(Node* newNode)
{
  if (!itsNext)
    itsNext = newNode;
  else
  {
    int NextCatsAge = itsNext->GetCat()->GetAge();
    int NewAge =  newNode->GetCat()->GetAge();
    int ThisNodeAge = itsCat->GetAge();

    if (  NewAge >= ThisNodeAge && NewAge < NextCatsAge  )
    {
      newNode->SetNext(itsNext);
      itsNext = newNode;
    }
    else
```

```
            itsNext->Insert(newNode);
        }
    }

    void Node::Display()
    {
      if (itsCat->GetAge() > 0)
      {
        cout << "My cat is ";
        cout << itsCat->GetAge() << " years old\n";
      }
      if (itsNext)
        itsNext->Display();
    }

    int main()
    {

      Node *pNode = 0;
       CAT * pCat = new CAT(0);
       int age;

       Node *pHead = new Node(pCat);

       while (1)
       {
         cout << "New Cat's age? (0 to quit): ";
         cin >>  age;
         if (!age)
           break;
        pCat = new CAT(age);
        pNode = new Node(pCat);
        pHead->Insert(pNode);
       }
       pHead->Display();
       delete pHead;
       cout << "Exiting...\n\n";
    return 0;
}
```

الوحدة الرابعة

الدوال الصديقة والعمليات

Friend Functions and Operators

في هذه الوحدة، عزيزي الطالب، سندرس الدوال والأصناف الصديقة، وكيفية تعريفها واستخدامها. ثم سندرس كيفية تعريف العمليات مثل(+ و * و -) بحيث نجعلها تتصرف كما نريد حسب نوعية العوامل. وفي القسم الأخير سوف نتحدث عن المجموعات

2.1 أهداف الوحدة

ينتظر منك، عزيزي الطالب، بعد تنفيذ البرامج أن تكون قادرا على أن:

1. تكتب وتستخدم الدوال التي تستقبل العوامل بالإشارة .Reference Parameters.
2. تكتب الدوال الصديقة وتستدعيها بشكل صحيح.
3. تعرف الأصناف الصديقة.
4. تكتب تعريفات جديدة للعمليات المختلفة.
5. تعرف وتستخدم this .

تدريب (1)

الطرق المختلفة لتمرير العوامل، اكتب البرامج التالية ونفذها على الحاسوب ، وعلل نتائج التنفيذ.

تمرير العوامل بالإشارة .parameter passing by reference	تمرير العوامل بالقيمة parameter passing by value.

void swap1(int X, int Y) { int T=X; X=Y; Y=T; }	void swap2(int* X,int* Y) { int T=*X; *X=*Y; *Y=T; }	void swap3(int &X, int &Y) { int T=X; X=Y; Y=X; }

main()
{int A=10;
int B=20;
 	
cout<<"A="<<A<<"
B="<<B<<endl;
swap1(A,B);
 	
cout<<"A="<<A<<"
B="<<B<<endl;
}
	main()
	{int A=10;
	int B=20;
	
	cout<<"A="<<A<<"
	B="<<B<<endl;
	swap2(&A,&B);
	
	cout<<"A=ط<<A<<"
	B="<<B<<endl;
	}
		main()
		{int A=10;
		int B=20;
		
		cout<<"A="<<A
		<<"
		
		B="<<B<<endl;
		swap3(A,B);
		
		cout<<"A="<<A
		<<"
		
		B="<<B<<endl;
		}

A=10 B=20 A=10 B=20	A=10 B=20 A=20 B=10	A=10 B=20 A=20 B=10

تدريب (2)

الدوال الصديقة Friend Functions

تعرف الدالة على أنها دالة صديقة لصنف ما بوضع نموذج prototype لهذه الدالة داخل تعريف الصنف مسبوقا بالكلمة friend . مما يعني أن الدالة الصديقة لها الحق في معالجة الأجزاء الخاصة للصنف كما لو كانت دالة منتمية للصنف. وتختلف طريقة استدعاء الدالة الصديقة عن طريقة استدعاء الدالة المنتمية.
ادرس البرنامج التالي جيدا، حيث يقوم البرنامج بتعريف صنف اسمه مسافة Distance لإيجاد مربع المسافة المعطاة بالقدم والبوصة بالقياس قدم مربع، ثم اجب عن الأسئلة التالية:

```
// friend square() function for Distance
#include <iostream>
////////////////////////////////////////////////////////////
class Distance                  //English Distance class
  {
  private:
    int feet;
    float inches;
  public:
    Distance() : feet(0), inches(0.0) //constructor (no args)
      {  }
                                //constructor (two args)
    Distance(int ft, float in) : feet(ft), inches(in)
      {  }
    void showdist()                 //display distance
      { cout << feet << "\'-" << inches << '\"'; }
    friend float square(Distance);    //friend function
  };
//-----------------------------------------------------------
float square(Distance d)          //return square of
  {                             //this Distance
  float fltfeet = d.feet + d.inches/12;  //convert to float
  float feetsqrd = fltfeet * fltfeet;    //find the square
  return feetsqrd;              //return square feet
  }
////////////////////////////////////////////////////////////
int main()
  {
  Distance dist(3, 6.0);        //two-arg constructor (3'-6")
  float sqft;

  sqft = square(dist);          //return square of dist
                              //display distance and square
  cout << "\nDistance = "; dist.showdist();
  cout << "\nSquare = " << sqft << " square feet\n";
  return 0;
  }
```

1. ما هي الدالة الصديقة من البرنامج .....

2. هل نستخدم اسم الصنف او عملية تقرير المجال :: في ترويسة هذه الدالة .....
3. كيف يتم استدعاء الدالة الصديقة (اكتب جملة الاستدعاء من البرنامج السابق) .....
4. كيف تستخدم المتغيرات المنتمية داخل الدوال الصديقة اعطي مثال على ذلك .....
5. نفذ البرنامج واكتب نتيجة التنفيذ .....

تدريب (3) الأصناف الصديقة

لتجنب وضع عدد كبير من الدوال المنتمية إلى صنف كدوال صديقة إلى صنف آخر، نستطيع تعريف الصنف بأكمله كصنف صديق للصنف الآخر. مما يعطي جميع الدوال المنتمية إلى الصنف الأول القدرة على معالجة الأجزاء الخاصة في الصنف الآخر. على سبيل المثال لنفرض أن لدينا الصنفين beta والصنف alpha وأننا نود إعطاء جميع الدوال المنتمية للصنف beta إمكانية معالجة جميع المتغيرات والدوال الخاصة بالصنف employee فبدلا من أن نعرف كل من هذه الدوال كدالة صديقة نستطيع تعريف الصنف beta كصديق للصنف alpha مما يحقق نفس الغاية.

ادرس البرنامج التالي جيدا ونفذة على الحاسوب واطبع نتيجة التنفيذ:

```
// friend classes
#include <iostream>
////////////////////////////////////////////////////////////
class alpha
  {
  private:
    int data1;
  public:
    alpha() : data1(99) { }   //constructor
    friend class beta;        //beta is a friend class
  };
////////////////////////////////////////////////////////////
class beta
  {                           //all member functions can
  public:                     //access private alpha data
    void func1(alpha a)  { cout << "\ndata1=" << a.data1; }
    void func2(alpha a)  { cout << "\ndata1=" << a.data1; }
  };
////////////////////////////////////////////////////////////
int main()
  {
  alpha a;
  beta b;

  b.func1(a);
  b.func2(a);
  cout << endl;
  return 0;
  }
```

تدريب (4)
إعادة تحميل عملية الإسناد المساواة
من المميزات الجميلة للغة C++ أنها تمكننا من إعادة تعريف العمليات operator المختلفة بما يتناسب مع تطبيقاتنا. فمثلا نستطيع تعريف العملية = كعملية منتمية للصنف alpha لتعني اسناد – لتعني اسناد كائن الى اخر.

```
// overloads assignment operator (=)
#include <iostream>
////////////////////////////////////////////////////////////
class alpha
   {
   private:
      int data;
   public:
      alpha()                    //no-arg constructor
         { }
      alpha(int d)               //one-arg constructor
         { data = d; }
      void display()             //display data
         { cout << data; }
      alpha operator = (alpha& a)  //overloaded = operator
         {
         data = a.data;          //not done automatically
         cout << "\nAssignment operator invoked";
         return alpha(data);     //return copy of this alpha
         }
   };
////////////////////////////////////////////////////////////
int main()
   {
   alpha a1(37);
   alpha a2;

   a2 = a1;                   //invoke overloaded =
   cout << "\na2="; a2.display();  //display a2

   alpha a3 = a2;             //does NOT invoke =
   cout << "\na3="; a3.display();  //display a3
   cout << endl;
   return 0;
   }
```

تدريب (5)
إعادة تحميل عملية الإسناد المساواة
استخدم فكرة إعادة التحميل لإسناد المساواة في التدريب 2

```
// friend overloaded + operator
#include <iostream>
////////////////////////////////////////////////////////////
class Distance                  //English Distance class
  {
  private:
    int feet;
    float inches;
  public:
    Distance()                  //constructor (no args)
      { feet = 0; inches = 0.0; }
    Distance( float fltfeet )   //constructor (one arg)
      {                         //convert float to Distance
      feet = int(fltfeet);          //feet is integer part
      inches = 12*(fltfeet-feet);   //inches is what's left
      }
    Distance(int ft, float in)  //constructor (two args)
      { feet = ft; inches = in; }
    void showdist()             //display distance
      { cout << feet << "\'-" << inches << '\"'; }
    friend Distance operator + (Distance, Distance); //friend
  };
//-----------------------------------------------------------
Distance operator + (Distance d1, Distance d2) //add D1 to d2
  {
  int f = d1.feet + d2.feet;        //add the feet
  float i = d1.inches + d2.inches;  //add the inches
  if(i >= 12.0)                 //if inches exceeds 12.0,
    { i -= 12.0; f++;  }        //less 12 inches, plus 1 foot
  return Distance(f,i);         //return new Distance with sum
  }
//-----------------------------------------------------------
int main()
  {
  Distance d1 = 2.5;                //constructor converts
  Distance d2 = 1.25;               //float-feet to Distance
  Distance d3;
  cout << "\nd1 = "; d1.showdist();
  cout << "\nd2 = "; d2.showdist();

  d3 = d1 + 10.0;                   //distance + float: OK
  cout << "\nd3 = "; d3.showdist();
```

```
  d3 = 10.0 + d1;                 //float + Distance: OK
  cout << "\nd3 = "; d3.showdist();
  cout << endl;
  return 0;
  }
```

تدريب (6)

في كلّ مرة يتم فيها استدعاء الدوال المنتمية ، فانه يمرر بصورة اتوماتيكية مؤشر يدعى this ، إلى الكينونة التي تم استدعائها. هذا المؤشر يدعى this ويعتبر معامل ضمني لجميع الدوال المنتمية. لذا ،في داخل اي دوال منتمية، المؤشر this قد يستعمل للإشارة إلى الكينونة المستدعاة.

نفذ البرنامج التالي وبين النتيجة التي حصلت عليها ، ونقاش النتيجة التي حصلت عليها

```
// this
#include <iostream>

class CDummy {
  public:
    int isitme (CDummy& param);
};

int CDummy::isitme (CDummy& param)
{
  if (&param == this) return true;
  else return false;
}

int main () {
  CDummy a;
  CDummy* b = &a;
  if ( b->isitme(a) )
    cout << "yes, &a is b";
  return 0;
}
```

الوحدة الخامسة

الأصناف المشتقة والقوالب

Derived Classes and Templates

في هذه الوحدة، عزيزي الدارس، سـنناقش أسـاليب مختلفة لتحقيق الهدف الأهم للغات البرمجة الكينونية وهو تقليل الحاجة إلى كتابة دوال جديدة عن طريق إعادة استخدام الدوال القديمة Code Reuse. إذ سـنناقش في هذه الوحدة كيفية اشـتقاق صـنف جديد، يدعى بالصـنف المشـتق Derived Class ، من صنف (أو أكثر) آخر يدعى بالصنف الأساس Base Class ؛ مما يسمح لنا، إذا أحسنا تصميم النظام، استخدام الكثير من الدوال الخاصـة بالصـنف الأساس للصـنف المشـتق دون الحاجة إلى إعادة كتابتها من جديد. وذلك لأن المتغيرات والدوال الخاصـة بالصـنف الأساس توّرث إلى الصـنف المشـتق. لاحظ، عزيزي الدارس، أن إعادة استخدام الدوال Code Reuse لا يقلل، فقط، من الجهد والوقت اللازمين لتطوير البرامج، إنما أيضـاً يقلل من الوقت والجهد اللازمين لصـيانتها. ويسـاعدنا أيضـاً على إنتاج برامج يعتمد عليها reliable، إذ إن استخدام دالة مجربة خير من كتابة دالة جديدة قد تحوي أخطاء خفية.

ومن الطرق الأخرى التي تزودنا بها لغة ++ C لتساعدنا في تحقيق هدفنا في إعادة استخدام البرامج هو ما يسمى بالدوال القالبية Template Functions وهي دوال تكتب بطريقة تسمح لنا باستخدامها على أنواع مختلفة من البيانات دون الحاجة إلى كتابة نسـخة من هذه الدوال لكل نوع من أنواع البيانات. كذلك تسـمح لنا لغة ++ C بتعريف أصـناف قالبية وهي أصـناف عامة تسـمح لنا بتأجيل تحديد نوع المتغيرات المنتمية إلى حين استخدامها، وبذلك فهي تمنحنا القدرة على تعريف أصناف خاصة تعمل على أنواع بيانات مختلفة دون الحاجة إلى كتابة هذه الأصناف المختلفة وتعريف دوالها.

وممن المميزات الهامة للغة ++ C ما يدعى بتعدد الأوجه polymorphism، حيث تسـمح لنا لغة ++ C بتأخير تحديد هوية الدالة التي ستستدعى إلى وقت تنفيذ البرنامج (وليس وقت ترجمته كما هي العادة)

مما يمنحنا القدرة على كتابة برامج تقوم بمهام من الصـــعب كتابتها بلغات البرمجة التقليدية أو ما يســـمى بلغات البرمجة الإجرائية كالباسكال، على سبيل المثال.

أهداف الوحدة

يتوقع منك، عزيزي الدارس، عند فراغك من دراسة هذه الوحدة، أن تكون قادراً على أن :

1. تصمم ما يسمى بشجرة الأصناف الهرمية بحيث تقلل الحاجة إلى إعادة كتابة دوال جديدة قدر الإمكان.
2. تعرف أصنافاً جديدة مشتقة من أصناف أخرى مفردة أو متعددة.
3. تبين أنواع الوراثة المختلفة وكيف تستخدمها.
4. تعرف دوال قالبية Template Functions وتستخدمها بطريقة صحيحة.
5. تعرف صنفاً قالبياً Template Class وتعرف كيف تستخدمه لعمل أصناف جديدة.
6. تعرف كيف ومتى تستخدم خاصية تعدد الأوجه polymorphism المهمة.

الوراثة المنفردة Single Inheritance:

لو فرضنا ان لدينا الصنف Shape والذي يعتبر الصنف الأساس لأي شكل هندسي فإنه يمكننا أن نكتب الصنف Rectangle والذي يمثل المستطيل بأن يكون مشتقا (أن يرث) من الصنف Shape، أدرس المثال التالي ونفذه على الحاسوب:

```
#include <iostream>
#include <conio.h>

// Base class
class Shape
{
  public:
    void setWidth(int w)
    {
      width = w;
    }
    void setHeight(int h)
    {
      height = h;
    }
  protected:
```

```
    int width;
    int height;
};

// Derived class
class Rectangle: public Shape
{
  public:
    int getArea()
    {
      return (width * height);
    }
};

int main(void)
{
  Rectangle Rect;
  Rect.setWidth(5);
  Rect.setHeight(7);

  // Print the area of the object.
  cout << "Total area: " << Rect.getArea() << endl;
getch();
}
```

ناتج تنفيذ المثال السابق هو:

```
Total area: 35
```

الوراثة المتعددة Multiple Inheritance:

أدرس المثال التالي والذي يوضح مفهوم الوراثة المتعددة ونفذه على الحاسوب.

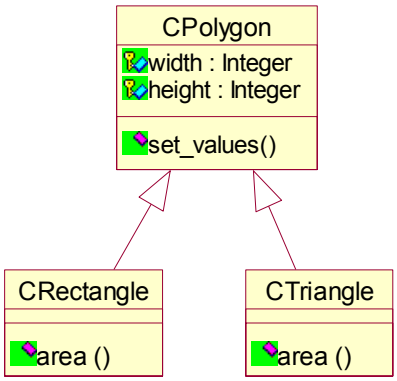
```
#include <iostream>
 #include <conio.h>

// Base class Shape
class Shape
{
  public:
    void setWidth(int w)
    {
      width = w;
    }
    void setHeight(int h)
    {
      height = h;
```

```
    }
  protected:
    int width;
    int height;
};

// Base class PaintCost
class PaintCost
{
  public:
    int getCost(int area)
    {
      return area * 70;
    }
};

// Derived class
class Rectangle: public Shape, public PaintCost
{
  public:
    int getArea()
    {
      return (width * height);
    }
};

int main(void)
{
  Rectangle Rect;
  int area;

  Rect.setWidth(5);
  Rect.setHeight(7);

  area = Rect.getArea();

  // Print the area of the object.
  cout << "Total area: " << Rect.getArea() << endl;

  // Print the total cost of painting
  cout << "Total paint cost: $" << Rect.getCost(area) << endl;

getch();
}
```

ناتج تنفيذ المثال السابق هو:

Total area: 35
Total paint cost: $2450

تمرين(1): اكتب النص البرمجي الذي يحقق الوراثة المنفردة الموضحة في الشكل التالي علما بان الصنفان المشتقان يحتويان دوال حساب المساحة.



مساعدة: النص البرمجي للصنف CPolygon هو:

```
// derived classes
#include <iostream>
#include <conio.h>
class CPolygon {
  protected:
        int width, height;
  public:
      void set_values (int a, int b)
                    { width=a; height=b;}
};
```

تمرين(2): لو عرّفنا صنفا أساسيا اسمه COutput وأردنا من الصنفين CTriangle, CRectangle أن يرثا من هذا الصنف بالإضافة إلى الوراثة من الصنف CPolygon السابق. اعمل التعديل اللازم على البرنامج السابق لتحقيق مفهوم الوراثة المتعددة.

مساعدة: تعريف الصنف COutput هو:

```
class COutput {
public:
```

```
void output (int i);
};
```

تعدد الاوجه Polymorphism

ادرس المثال التالي والذي يبين مفهوم الوراثة المتعددة ونفذه باستخدام مترجم C++. لاحظ ناتج التنفيذ ثم أجب عن السؤال الذي يليه.

```
#include <iostream>
#include <conio.h>
class Shape {
  protected:
    int width, height;
  public:
    Shape( int a=0, int b=0)
    {
      width = a;
      height = b;
    }
    int area()
    {
      cout << "Parent class area :" <<endl;
      return 0;
    }
};
class Rectangle: public Shape{
  public:
    Rectangle( int a=0, int b=0)
    {
     Shape(a, b);
    }
    int area ()
    {
      cout << "Rectangle class area :" <<endl;
      return (width * height);
    }
};
class Triangle: public Shape{
  public:
    Triangle( int a=0, int b=0)
    {
     Shape(a, b);
    }
    int area ()
```

```
    {
      cout << "Rectangle class area :" <<endl;
      return (width * height / 2);
    }
};
// Main function for the program
int main( )
{
  Shape *shape;
  Rectangle rec(10,7);
  Triangle  tri(10,5);

  // store the address of Rectangle
  shape = &rec;
  // call rectangle area.
  shape->area();

  // store the address of Triangle
  shape = &tri;
  // call triangle area.
  shape->area();
  getch();
}
```

ناتج تنفيذ المثال السابق هو

```
Parent class area
Parent class area
```

تمرين (3): أعمل التعديل اللازم على المثال السابق ليحقق مفهوم تعدد الاوجه Polymorphism ويعطي النتيجة التالية:

```
Rectangle class area
Triangle class area
```

القوالب:

المثال التالي يعرف قالبا للدالة Swap والتي تعمل على تبديل متغيران من نوع كائن من نفس الصنف.

```
template <class Type>
void Swap (Type &x, Type &y)
{
        Type tmp = x;
        x = y;
        y = tmp;
}
```

تمرين(4): اكتب قالبا للدالة GetMax والتي تعيد أكبر كائن من الكائنين الممررين إليها، ثم استدع هذه الدالة مرتين، الأولى لمقارنة عددين صحيحين ، ومرة أخرى لتقارن عددين من نوع float.

تمرين(5): اكتب القالب GetMin والذي يقبل نوعين مختلفين من البيانات. وقم بتنفيذه.

تمرين(6): اكتب صنفا يقبل أعضاء من أنواع عامة ويسطيع أن يخزن عنصرين من أي نوع مقبول ويحتوي دالة تعيد أكبر قيمة.

حلول تمرينات الوحدة الخامسة

حل التمرين (1):

```
// derived classes
#include <iostream>
#include <conio.h>
class CPolygon {
protected:
int width, height;
public:
void set_values (int a, int b)
{ width=a; height=b;}
};

class CRectangle: public CPolygon {
public:
int area (void)
{ return (width * height); }
};
class CTriangle: public CPolygon {
public:
int area (void)
{ return (width * height / 2); }
};
int main () {
CRectangle rect;
CTriangle trgl;
rect.set_values (4,5);
trgl.set_values (4,5);
cout << rect.area() << endl;
cout << trgl.area() << endl;
getch();
}
```

حل التمرين (2):

```
// multiple inheritance
#include <iostream>
#include <conio.h>
class CPolygon {
protected:
```

```
int width, height;
public:
void set_values (int a, int b)
{ width=a; height=b;}
};
class COutput {
public:
void output (int i);
};
void COutput::output (int i) {
cout << i << endl;
}
class CRectangle: public CPolygon, public
COutput {
public:
int area (void)
{ return (width * height); }
};
class CTriangle: public CPolygon, public
COutput {
public:
int area (void)
{ return (width * height / 2); }
};
int main () {
CRectangle rect;
CTriangle trgl;
rect.set_values (4,5);
trgl.set_values (4,5);
rect.output (rect.area());
trgl.output (trgl.area());
getch();
}
```

حل التمرين (3): التعديل هو على تعريف الدالة area في الصنف Shape بأن يسبق تعريفها الكلمة virtual من أجل تحقيق مفهوم تعدد الأشكال Polymorphism.

```
class Shape {
  protected:
    int width, height;
  public:
    Shape( int a=0, int b=0)
    {
```

```
        width = a;
        height = b;
    }
    virtual int area()
    {
        cout << "Parent class area :" <<endl;
        return 0;
    }
};
```

حل التمرين (4):

```
// function template
#include <iostream>
#include <conio.h>
template <class S>
S GetMax (S a, S b) {
S result;
return ( result = (a>b)? a : b );
}
int main () {
int i=5, j=6, k;
float l=10.0, m=5.0, n;
k=GetMax<int>(i,j);
n=GetMax<float>(l,m);
cout << k << "\n" << n << endl;
getch();
}
```

حل التمرين (5):

```
// function template
#include <iostream>
#include <conio.h>
template <class T, class U>
T GetMin (T a, U b) {
return ( a < b ? a : b ) ;
}
int main () {
int i=5, j=6, k;
float l=10.0, m=5.0, n;
k = GetMin<int,float> (j,l);
n = GetMin (j,l);
cout << k << "\n" << n << endl;
getch();
```

```
}
```

حل التمرين (6):

```
// class templates
#include <iostream>
#include <conio.h>
template <class T>
class mypair {
T a, b;
public:
mypair (T first, T second)
{a=first; b=second;}
T getmax ();
};
template <class T>
T mypair<T>::getmax ()
{
T retval;
retval = a>b? a : b;
return retval;
}
int main () {
mypair <int> myobject (100, 75);
cout << myobject.getmax()<<endl;
getch();
}
```

الوحدة السادسة

معالجة الملفات بلغة C++

File Processing with C++

عزيزي الدارس، تهدف هذه الوحدة إلى إعطائك معلومات أساسية عمّا يتعلق بمعالجة الملفات في لغة C ++ وتوضيح بعض العمليات الأساسية على الملفات بأنواعها المختلفة.

تتناول هذه الوحدة تعريف مفهوم هيكل البيانات وموقع الملفات بينها وإعطاء لمحة مختصرة عن الأنواع المختلفة من البيانات. ويتلو ذلك مناقشة الطرق المختلفة في تنظيم الملفات وتعريف الملفات والينابيع، ومن ثم توضيح كيفية إنشاء الملفات التتابعية والعشوائية وقراءتها وتحديثها، وأخيراً تم توضيح كيفية التعامل مع الكائنات من حيث الإدخال والإخراج في الملفات.

أهداف الوحدة

ينتظر منك، عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة، أن تكون قادراً على أن:

1. تعرّف هيكل البيانات وموقع الملفات بينها.
2. تعرّف الطرق المختلفة في تنظيم الملفات (File Organization).
3. تعرّف الملفات والينابيع (Files and Streams)
4. تنشئ الملفات التتابعية وتقرؤها وتحدثها.
5. تنشئ الملفات العشوائية وتقرؤها وتحدثها.
6. تدخل الكائنات وتخرجها (Input/Output of Objects).

المثال التالي يوضح كيفية القراءة والكتابة في ملف، أدرس المثال المكتوب ونفذه على الحاسوب ثم قم بما يلي:

1- أدخل البيانات المطلوبة أثناء التنفيذ.

2- أنظر ناتج التنفيذ على الشاشة (سيظهر من ضمنها المعلومات التي قمت بإدخالها).

3- بعد تنفيذ البرنامج سيتم انشاء ملف باسم afile.dat على مجلد العمل.

4-افتح الملف afile.dat بواسطة محرر نصوص مثل برنامج notepad مثلا، ما محتويات الملف؟

```
#include <fstream>
#include <iostream>
#include <conio.h>
int main ()
{
  char data[100];
  // open a file in write mode.
  ofstream outfile;
  outfile.open("afile.dat");

  cout << "Writing to the file" << endl;
  cout << "Enter your name: ";
  cin.getline(data, 100);

  // write inputted data into the file.
  outfile << data << endl;

  cout << "Enter your age: ";
  cin >> data;
  cin.ignore();

  // again write inputted data into the file.
  outfile << data << endl;

  // close the opened file.
  outfile.close();

  // open a file in read mode.
  ifstream infile;
  infile.open("afile.dat");

  cout << "Reading from the file" << endl;
  infile >> data;
```

```
  // write the data at the screen.
  cout << data << endl;

  // again read the data from the file and display it.
  infile >> data;
  cout << data << endl;

  // close the opened file.
  infile.close();

  getch();
}
```

تمرين (1): اكتب برنامجا لكتابة نصا في سطرين داخل ملف example.txt. موضحا كيفية فتح الملف للكتابة دون مسح محتويات الملف إن كان موجودا من قبل.

تمرين (2): اكتب برنامجا يقوم بقراءة محتويات الملف المذكور في التمرين السابق وطباعة محتوياته على الشاشة. وإن كان الملف غير موجود فإنه يطبع رسالة تبين ذلك.

تمرين (3): اكتب برنامجا لحساب حجم الملف example.txt

تمرين (4): اكتب برنامجا يقوم بكتابة عدد من سجلات طلبة في ملف عشوائي، حيث يتوقف البرنامج عندما يقوم المستخدم بإدخال رقم طالب أقل من 0 أو أكبر من 100 ، وهي مجموعة أرقام الطلبة المراد انشاء سجلات لهم.

ملاحظة: بعد كتابة البرنامج وتنفيذه بنجاح، حاول فتح الملف المنشأ بواسطة محرر نصوص ثم لاحظ كيف تظهر محتويات الملف!!

تمرين (5): اكتب برنامجا يقوم بقراءة عدد من السجلات من ملف التمرين السابق باستخدام الدالة read والدالة seekg .

حلول تمرينات الوحدة السادسة

حل تمرين (1):

```
// writing on a text file
#include <iostream>
#include <fstream>
int main () {
  ofstream myfile ("example.txt", ios::app);
  if (myfile.is_open())
  {
    myfile << "This is a line.\n";
    myfile << "This is another line.\n";
    myfile.close();
  }
  else cout << "Unable to open file";
  return 0;
}
```

حل التمرين (2):

```
// reading a text file
#include <iostream>
#include <fstream>
#include <string>

int main () {
  string line;
  ifstream myfile ("example.txt");
  if (myfile.is_open())
  {
    while ( myfile.good() )
    {
      getline (myfile,line);
      cout << line << endl;
    }
    myfile.close();
  }

  else cout << "Unable to open file";
  return 0;
}
```

حل التمرين (3):

```
// obtaining file size
#include <iostream>
#include <fstream>

int main () {
  long begin, end;
  ifstream myfile ("example.txt");
  begin = myfile.tellg();
  myfile.seekg (0, ios::end);
  end = myfile.tellg();
  myfile.close();
  cout << "size is: " << (end-begin) << " bytes.\n";
  return 0;
}
```

حل تمرين (4):

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <string.h>

//using namespace std;
const int No_Of_Records =100;
struct student
{
   int no;
   char name[32];
   float average;
 };

int main(void)
{
 student studentrecord;
int sno;
char sname[32];
float saverage;
ofstream outStudentFile("studrf.dat",ios::in|ios::out|ios::binary);
if(!outStudentFile)
{
cerr << "File could not be opened" << endl;
exit(1);
}
cout <<"\nEnter student no (1 to 100), any number else to end input: ";
```

```
cin >> sno;
while(sno >0 && sno<=No_Of_Records)
{
cout << "\nEnter student name: ";
cin >> sname;
cout << "\nEnter student average: ";
cin >> saverage;
studentrecord.no=sno;
strcpy(studentrecord.name,sname);
studentrecord.average=saverage;
outStudentFile.seekp((studentrecord.no-1)* sizeof(struct student));
outStudentFile.write(reinterpret_cast <const char *>
(&studentrecord),sizeof(struct student));
cout <<"\n\nEnter student no (1 to 100), any number else to end input: ";
cin >> sno;
} // end while
outStudentFile.close();
return 0;
}
```

حل التمرين (5):

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <string.h>
const int No_Of_Records =100;
int main(void)
{
struct student
{
int no;
char name[32];
float average;
};
student studentrecord;
int sno;
ifstream inStudentFile("studrf.dat",ios::in);
if(!inStudentFile)
{
cerr << "File could not be opened" << endl;
exit(1);
}
cout <<"\nEnter student no (1 to 100), any number else to end displaying records: ";
cin >> sno;
```

```
while (sno >0 && sno<=No_Of_Records)
{
inStudentFile.seekg((sno-1)* sizeof(struct student));
inStudentFile.read(reinterpret_cast <char *>
(&studentrecord),sizeof(struct student));
cout << "\nStudent Data for student no " << studentrecord.no;
cout << "\nName: " << studentrecord.name;
cout << "\nAverage: " << studentrecord.average;
cout <<"\n\nEnter student no (1 to 100), any number else to end displaying records: ";
cin >> sno;
} // end while
inStudentFile.close();
return 0;
}
```

الملاحق

إعداد بيئة التطوير
Preparing Integrated Development Environment

مقدمة:

عزيزي الطالب، إن أهم ما تحتاجه قبل الدخول في عالم البرمجة هو: برنامج تحرير النصوص: لتكتب فيه برنامج بلغة الـ C أو الـ C++، المصنف من أن يحوّل المصدر للغة مزدوجة. المكتشف و الذي يحدد لك أخطائك التي تقع فيها أثناء كتابتك للبرنامج. يمكننا - عزيزي الطالب – أن نحصل على برنامج "3 في واحد" يقوم بالأعمال الثلاثة في آن واحد، و هذا النوع من البرامج نسميها الـ IDE أو ما يعرف **ببيئات التطوير.**

عزيزي الطالب، توجد الكثير من بيئات التطوير(IDEs) المعتمدة ومنها برنامج Code::Blocks و برنامج Visual C++ Express ويعتبران من البرامج المجانية، كما يوجد برامج أخرى مثل Borland C++ وبرنامج Visual C++ وأخرى عديدة غير مجانية.

في هذه الوحدة – عزيزي الطالب – سنتطرق إلى شرح كيفية إعداد واستخدام بيئات التطوير التالية:
- Code::Blocks (مجاني ومفتوح المصدر)
- Visual C++ من شركة ميكروسوفت وهو غير مجاني، ولكن الشركة أصدرت نسخة مجانية منه اسمها Visual C++ Express.

القسم الأول: تحميل وتنصيب وتشغيل برنامج Code::Blocks

في هذا القسم –عزيزي الطالب- إن شاء الله سنتطرق لشرح مصور لبرنامج Code::Blocks . البعض يعتبره البرنامج الاكثر سهولة في الاستخدام والسلاسة، كما أنه يعمل جيدا على وندوز7 على عكس بعض من البرامج الأخرى.
برنامج الـ Code::Blocks: هو IDE ملائم جدا، مجاني و يعمل على الـWindows، و الـMac OS، والـLinux. لغته المتوفرة حاليا هي الانجليزية.

يحتوي هذا القسم على المواضيع التالية:
- تحميل ملف التنصيب.
- شرح عملية تنصيب البرنامج
- شرح واجهة البرنامج
- تشغيل البرنامج من خلال إنشاء مشروع جديد و تطبيق مثال بسيط

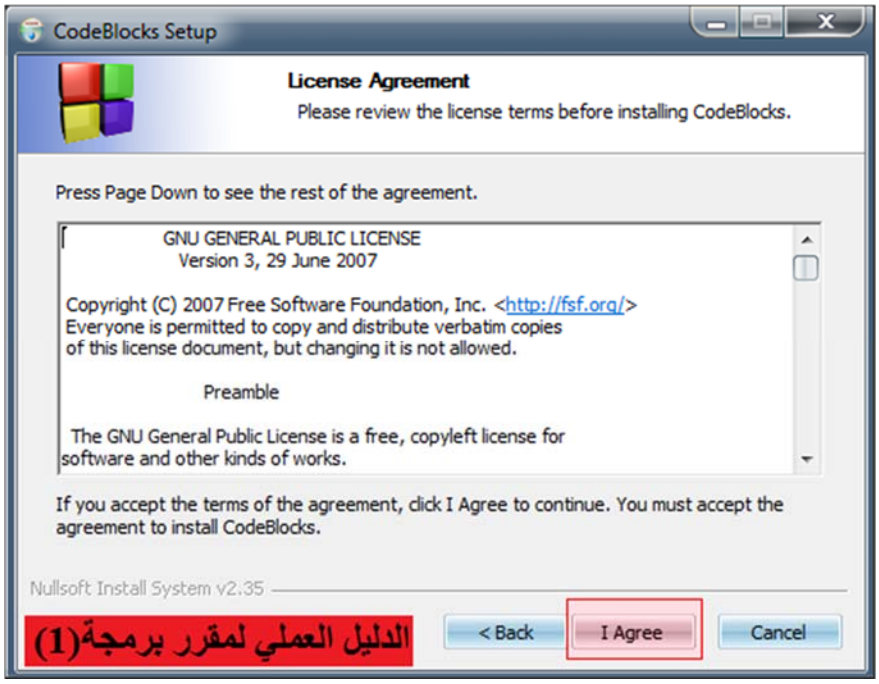
1. تحميل ملف التنصيب:

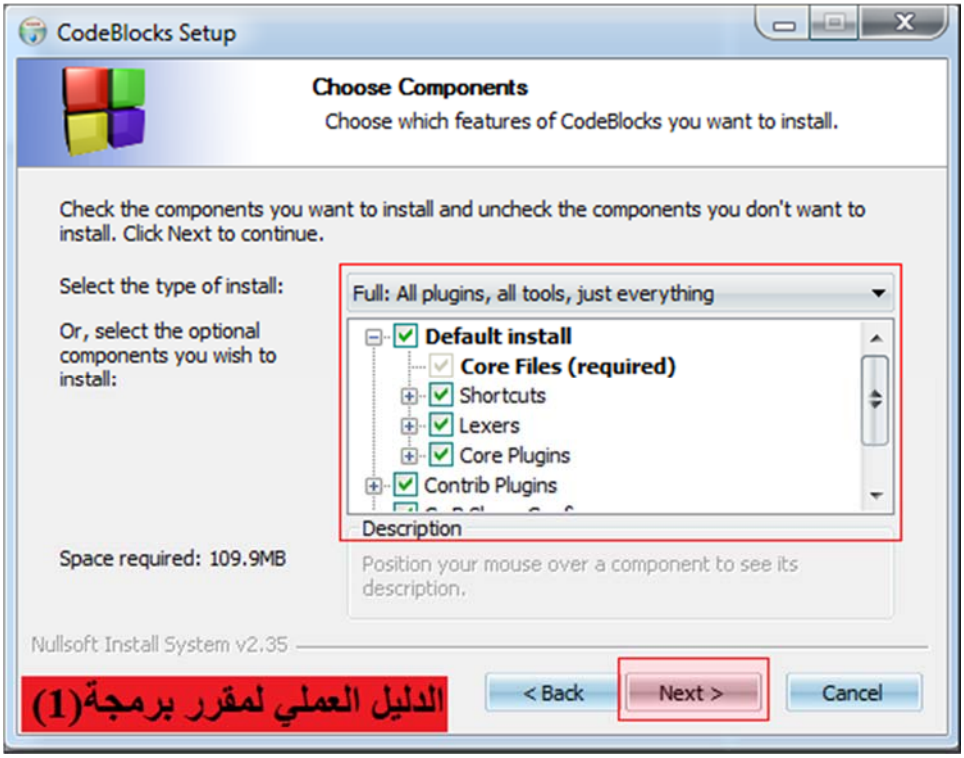
يمكنك تحميل ملف تنصيب البرنامج إلى جهازك من خلال النقر على الرابط التالي:
http://www.codeblocks.org/downloads/binaries

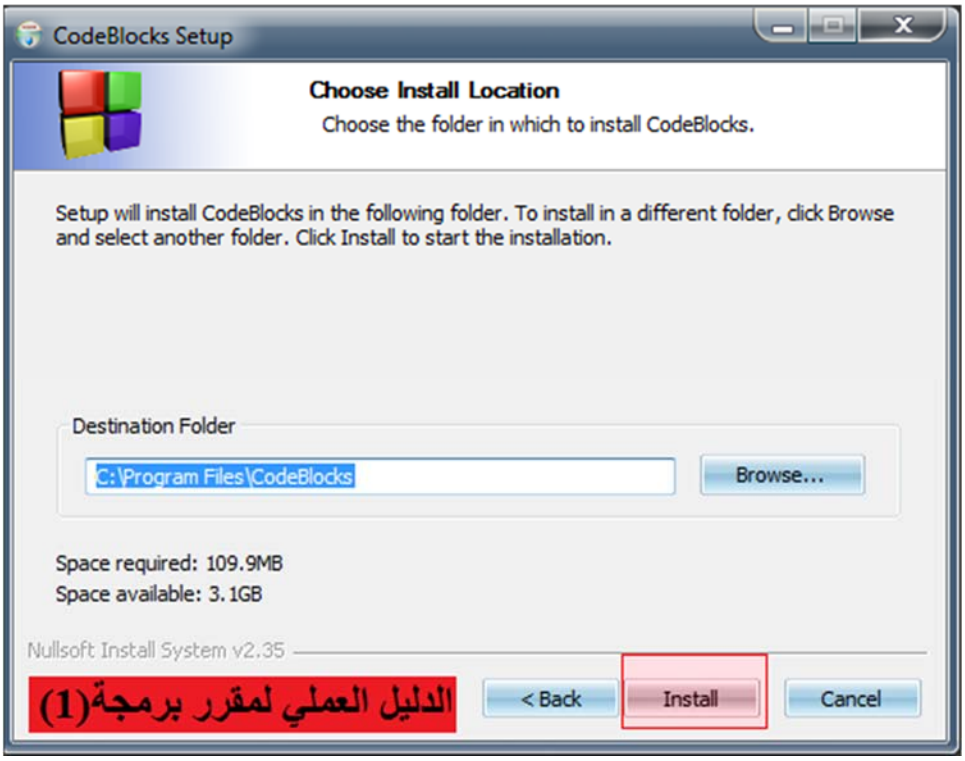
2. شرح عملية التنصيب:

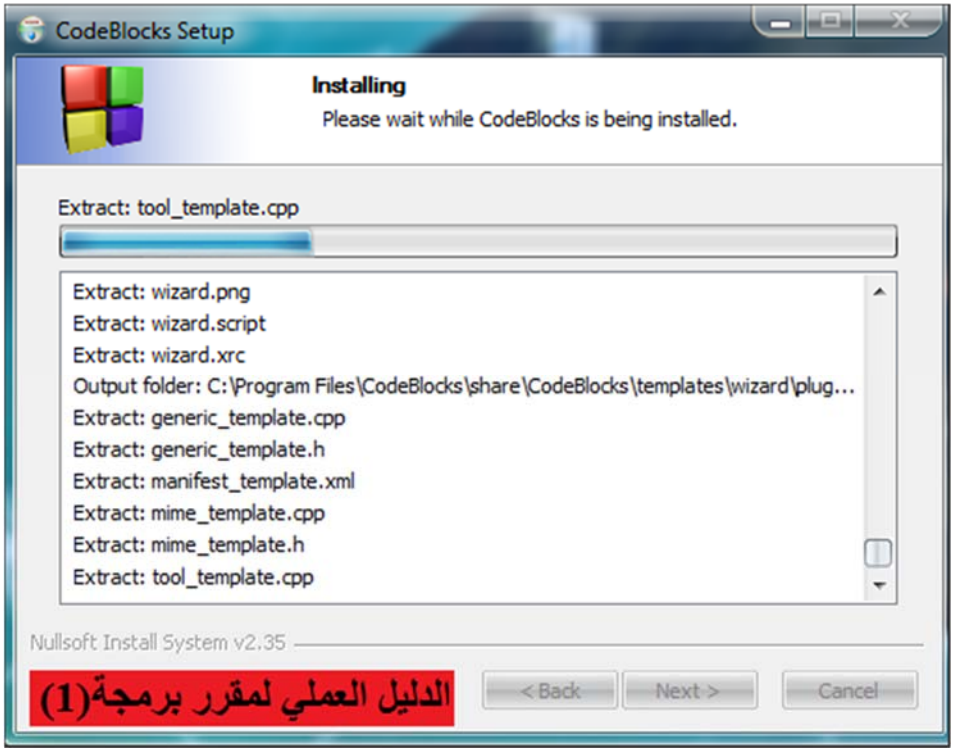
بعد تحميل ملف التنصيب، انقر عليه لتشغيله.

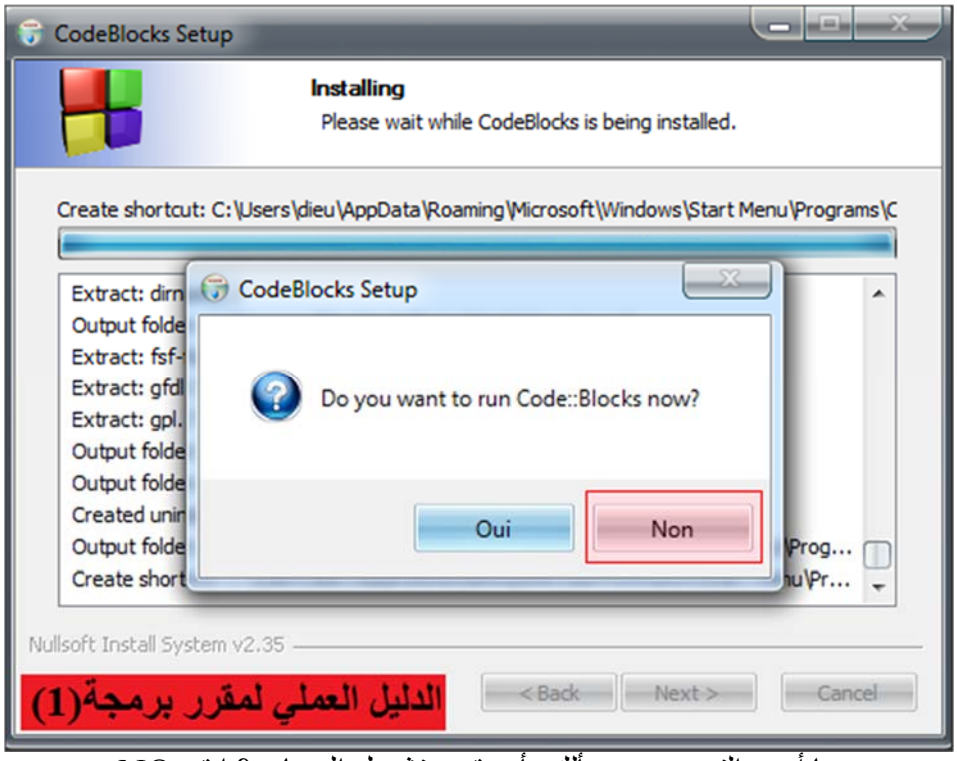
صورة 1: الملف الذي قمنا بتحميله


CodeBlocks Setup
License Agreement
Please review the license terms before installing CodeBlocks.
Press Page Down to see the rest of the agreement.
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The GNU General Public License is a free, copyleft license for
software and other kinds of works.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install CodeBlocks.
Nullsoft Install System v2.35
الدليل العملي لمقرر برمجة(1)
< Back
I Agree
Cancel


CodeBlocks Setup
Choose Components
Choose which features of CodeBlocks you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select the type of install:
Full: All plugins, all tools, just everything
Or, select the optional components you wish to install:
Default install
Core Files (required)
Shortcuts
Lexers
Core Plugins
Contrib Plugins
Description
Position your mouse over a component to see its description.
Space required: 109.9MB
Nullsoft Install System v2.35
الدليل العملي لمقرر برمجة(1)
< Back
Next >
Cancel


CodeBlocks Setup
Choose Install Location
Choose the folder in which to install CodeBlocks.
Setup will install CodeBlocks in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files\CodeBlocks
Browse...
Space required: 109.9MB
Space available: 3.1GB
Nullsoft Install System v2.35
الدليل العملي لمقرر برمجة(1)
< Back
Install
Cancel

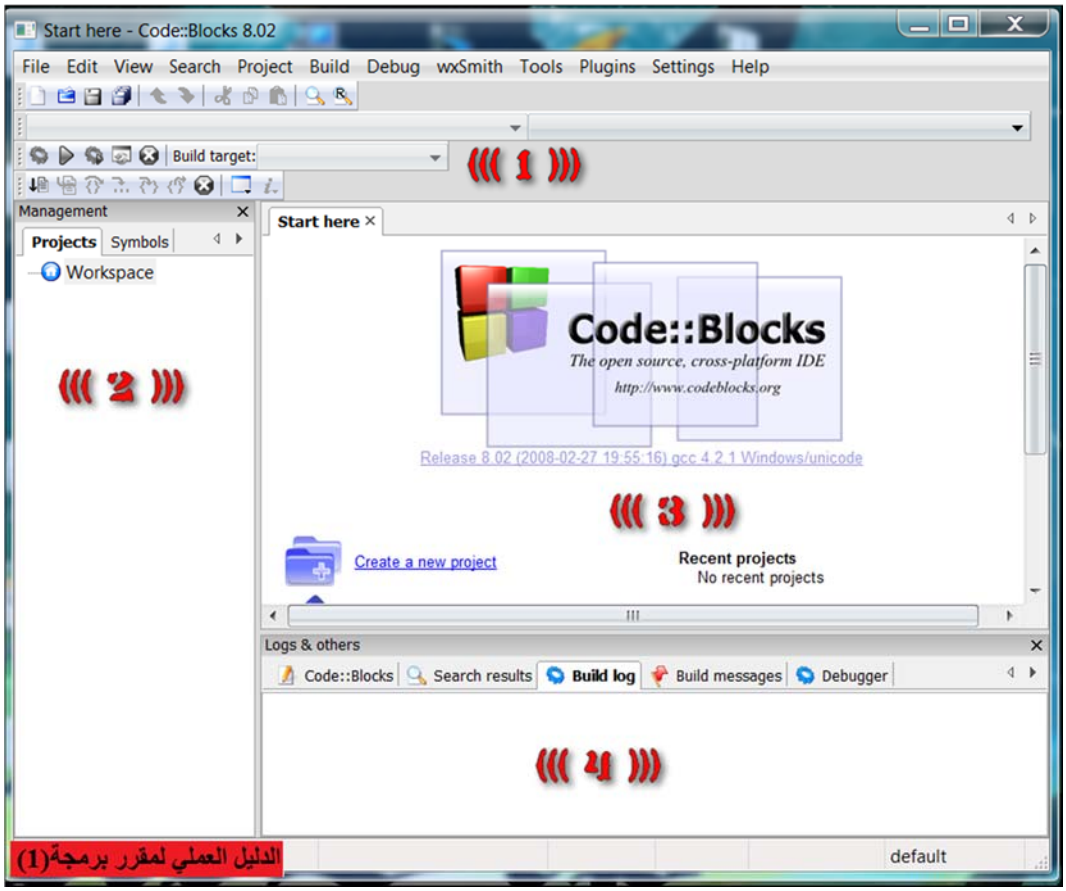

CodeBlocks Setup
Installing
Please wait while CodeBlocks is being installed.
Extract: tool_template.cpp
Extract: wizard.png
Extract: wizard.script
Extract: wizard.xrc
Output folder: C:\Program Files\CodeBlocks\share\CodeBlocks\templates\wizard\plug...
Extract: generic_template.cpp
Extract: generic_template.h
Extract: manifest_template.xml
Extract: mime_template.cpp
Extract: mime_template.h
Extract: tool_template.cpp
Nullsoft Install System v2.35
الدليل العملي لمقرر برمجة(1)
< Back
Next >
Cancel



هنا أنهى التنصيب ويسألك بأن يقوم بتشغيل البرنامج؟ انقر NO.

تم تنصيب البرنامج بنجاح، نقوم بفتح البرنامج.
ستجده في قائمة ابدأ.

3. شرح واجهة التطبيق



شرح دلالات الارقام:

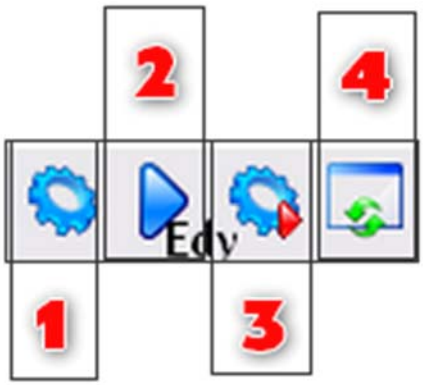
1- شريط الأدوات

2- قائمة ملفات المشروع : توجد بيسار النافذة، تحتوى على كل ملفات السورس المتعلقة بالبرنامج الذي تعمل عليه. و بما أنها فارغة حسب ما يظهر في الصورة هذا دليل على عدم وجود مصادر سورس في اللائحة حتى هذه اللحظة.

3- المنطقة الرئيسية : هنا المساحة التي تكتب فيها الكودسورس للغتك الـ C++.

4- منطقة البيان: و يسميها المبرمجون منطقة الموت، و هي المنطقة التي تعرض لك الأخطاء التي وقعت فيها أثناء كتابتك للكود سورس، طبعا متسلسلة بانتظام.

ما يهمنا الآن هو منطقة معينة خاصة من شريط الأدوات



1-Compile :

كل الكود سورس الخاصة ببرنامجكم ستبعث لمصنف يقوم بالترجمة ثم يستخرج لنا الـ Code::Blocks ملفا قابل للتنفيذ أو بالأحرى ملف تنفيذي. لكن إن اقترفت خطأ في الكتابة لن يترجم لك المصنف شيئا و ستظهر الأخطاء في منطقة البيان.

2- Execute :

تقوم بتشغيل آخر ملف تنفيذي قمت بكتابة الكود السورس الخاص به. و هذا عندما تريد تجريب البرنامج قبل الانتهاء منه. وستعرف بنفسك أنه قبل التشغيل Execute يجب الترجمة Compile ولكى نجرب ما توصلنا إليه نضغط على الأيقونة 3.

3- Compile & Execute:

لن تحتاج عبقرية زائدة لتعرف ماذا يمثل جمع وظيفتي الأيقونتين السابقتين، أضف لمعلوماتك -عزيزي الطالب- أنها الأيقونة التي سنكثر من استخدامها أكثر من باقيها، و أكرر ثانية بأنه إذا اقترفت خطأ في وضع الكود سورس لن يتم ترجمة الأكواد و لا تشغيل البرنامج! بل سيكون عليك تصحيح قائمة من الأكواد التي سبق و أخطأت فيها.

4- Recompile everything:

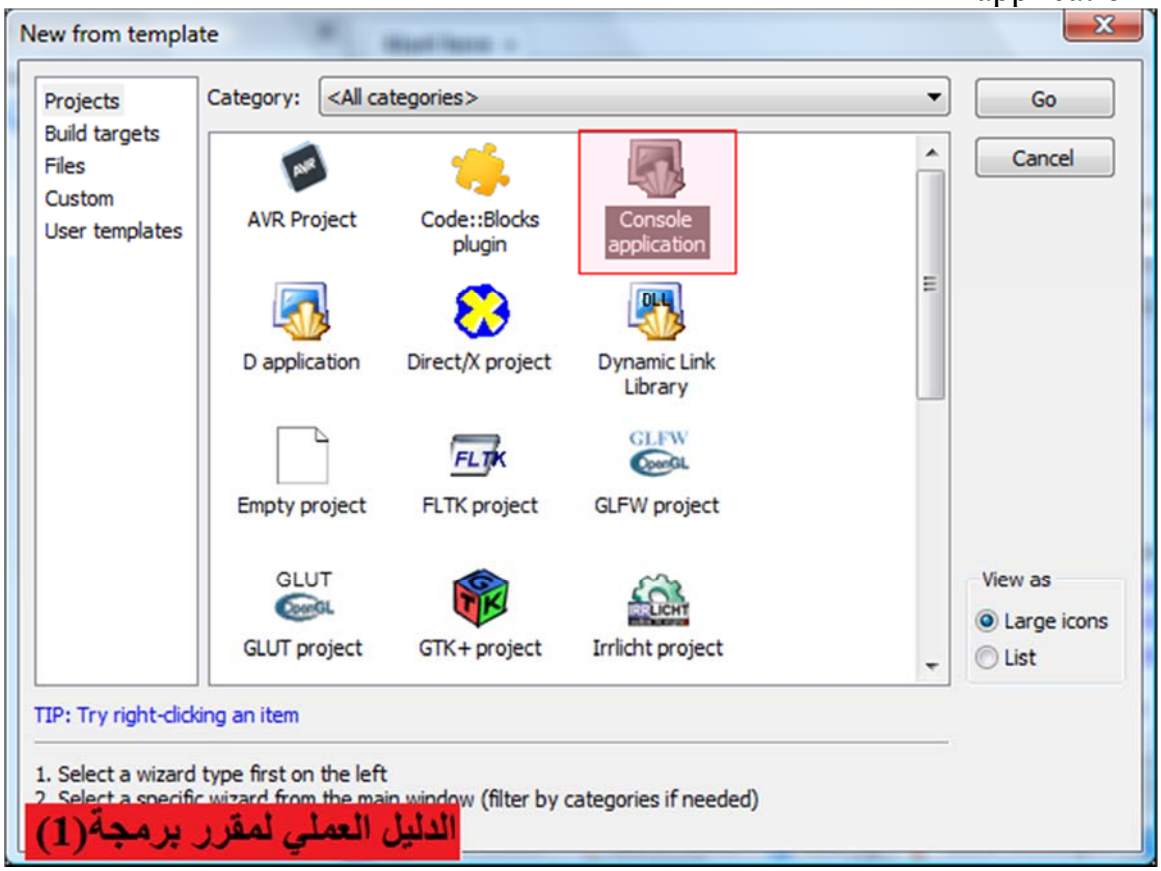
عندما نقوم بالـ"ترجمة" فإن Code::Blocks لن يقوم بتشغيل سوى الملفات التي عّدلت عليها أو أضفتها. أحيانا فقط، تضطر بأن تأمر الـ Code::Blocks بترجمة و تشغيل كل شيء Recompile everything.

الآن سنتطرق للمرحلة الأهم في البرنامج وهي تشغيل البرنامج.

4. تشغيل برنامج Code blocks: بداية مشروع جديد

على المستخدم انشاء مشروع جديد خاص به
الشرح كالتالي:

كي تبدأ مشروعا جديدا، توجه نحو File / New / Project : ، ستظهر لك نافذة اختر منها "Console application"



كما تلاحظ عزيزي الطالب، برنامج الـ Code::Blocks يسمح بالعمل على أنماط كثيرة من البرامج المختلفة، منها التي تعمل بمكتبات مشهورة كالـ SDL للـ 2D و ألـ OpenGL للـ 3D و الـ Qt و wxWidgets للـ نوافذ .. إلخ، هذه المكتبات غير مثبتة على جهازكم بعد، و بهذا لا يمكنكم جعلها تعمل.

سنبدأ أوّلا بالـ "Console":

أنقر على "Go" للشروع في المشروع الجديد.

بعدها سيأتيك اختيار بين لغتي الـ C أو الـ C++، اختر الـ C++.

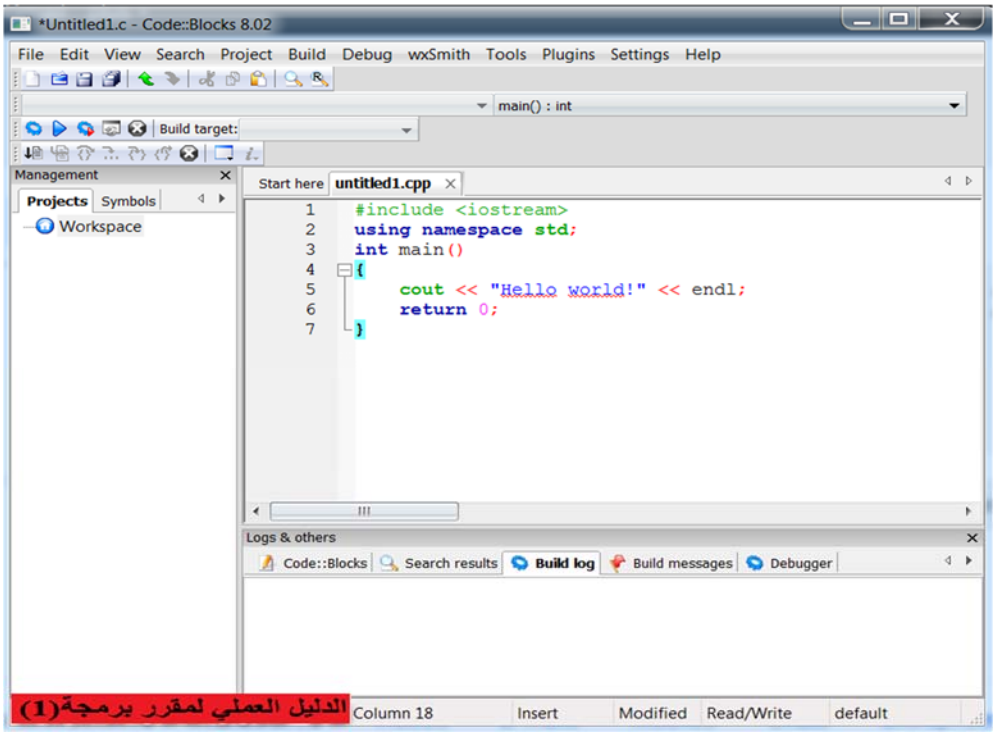
سيطلب منك الآن أن تدخل اسم المشروع، و كذا المسار الذي تختاره كي يحفظ فيه.

آخر خطوة تطلب منك هي ، كيف ينبغى أن يترجم البرنامج، يمكنك ترك الاختيارات على حالها، لن يكون لهذا أي تأثير على ما سنقوم به الآن، قم بتحديد زر. "Release"

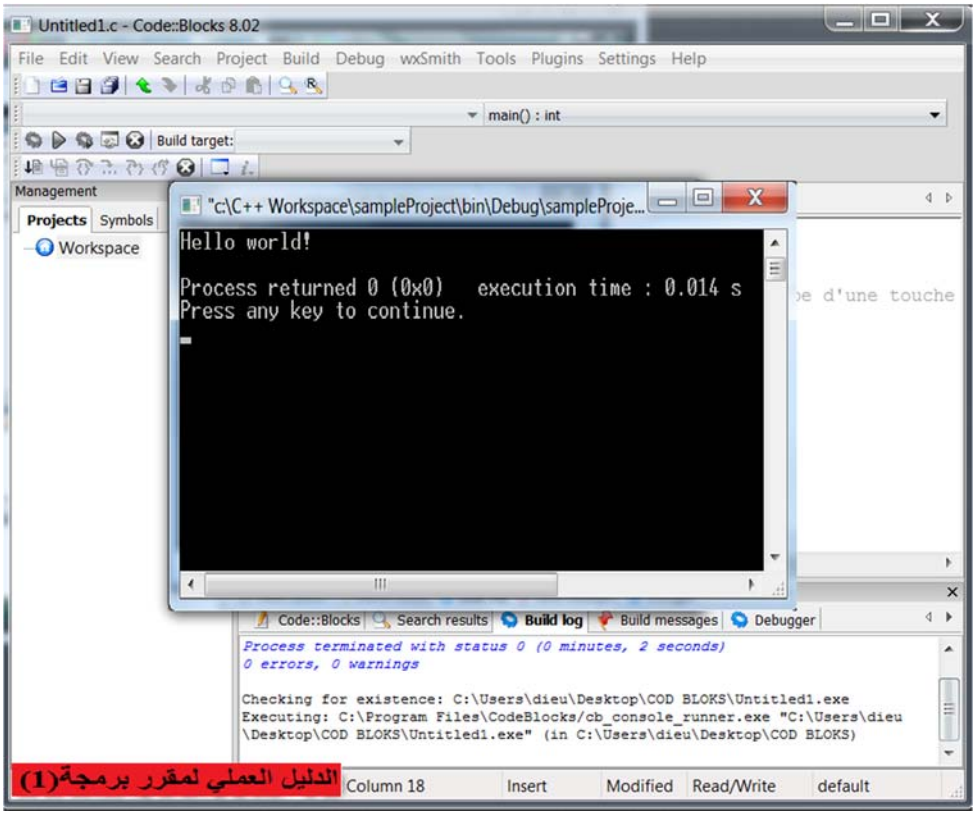
إظغط على Finish ، إنتهى ! . ماالذي قمنا به الآن يا ترى ؟
لقد قام البرنامج Code::Blocks بفتح مشروع جديد لنا مع احتواءه بعض الكود سورس التي سنحتاجها.

في الخانة الخاصة بالمشاريع على اليسار، إضغط على '+' لتوسيعها قليلا، و تظهر قائمة الملفات في المشروع.
الآن تم فتح مشروع خاص بك وما عليك الا بكتابة الكود والباقي على البرنامج،
سيقوم بفتح ملف تنفيذي يفتح بواسطة الMsDos :

تشغيل مثال بسيط:



النتيجة:



القسم الثاني: تشغيل برنامج Microsoft Visual C++

يمكنك عزيزي الطالب تنفيذ برامج لغة سي بلص بلص باستخدام العديد من المترجمات مثل: Borland ، Turbo c++، Microsoft Visual Studio 6.0 مايكروسوفت فيجيوال ستوديو نمط 6 او احدث من ذلك.

مايكروسوفت فيجيوال ستوديو Microsoft Visual Studio

مايكروسوفت فيجيوال ستوديو (بالإنجليزية Microsoft Visual Studio) : هي بيئة التطوير المتكاملة الرئيسية من مايكروسوفت. تتيح برمجة واجهة المستخدم الرسومية والبرامج النصية إلى جانب ويندوز فورم ومواقع ويب وتطبيقات ويب وخدمات الويب مدعومة ب مايكروسوفت ويندوز وويندوز موبايل وإطار عمل دوت نت ومايكروسوفت سيلفرلايت.

يحتوى فيجوال استوديو على محرر أكواد يدعم تقنية انتليسنس واعادة كتابة الكود, ويحتوى أيضا على مترجم يكشف أخطاء وقت التشغيل ومفسر يكشف الأخطاء الاملائية في الأكواد ويحتوى أيضا على مصمم نماذج لبناء واجهة مستخدم رسومية وغيرها.

يدعم فيجوال استوديو العديد من لغات البرمجة مثل مايكروسوفت فيجوال سي++ ومايكروسوفت فيجوال سي# ومايكروسوفت فيجوال بيسك وجافا سكريبت والعديد أيضا من لغات الترميز مثل html و xml و xhtml و xsl .

محرر الأكواد

مثل باقى بيئات التطوير يحتوى الفيجوال استوديو على محرر أكواد يدعم تعليم الصيغة والاكمال التلقائى ويستخدم أيضا تقنية انتليسنس لتساعد المبرمج في كتابة المتغيرات والدوال والوسائل والدورات بسرعة، ويدعمها المحرر في كتابة جميع لغات البرمجة والترميز التي يحتوى عليها الفيجوال استوديو.

يدعم محرر أكواد فيجوال استوديو أيضا إمكانية وضع علامات مرجعية في الكود للمساعدة في التصفح السريع، يدعم أيضا إمكانية collapsing and expanding مجموعات الأكواد والبحث والاستبدال في الكود, ويدعم أيضا code snippets وهى قوالب جاهزة من الكود يكمن ادراجها وادخال تغييرات عليها داخل المشروعات الجارى العمل عليها ويدعم أيضا إمكانية اعادة كتابة الكود.

يقوم محرر الأكواد أيضا بوضع خطوط حمراء أسفل الأخطاء الاملائية وخطوط خضراء أسفل الانذرات.

متعقب الأخطاء

يحتوى فيجوال ستوديو على متعقب أخطاء تدعمه جميع اللغات المدعومة يكشف أخطاء وقت التشغيل والأخطاء الاملائية ويسمح بوضع نقاط توقف عند سطور الكود والتي يتوقف البرنامج عن العمل عندما يصل لهذا السطر.
يوجد أيضا في فيجوال ستوديو نافذة immediate window والتي تسمح بتجريب الدوال أثناء كتابتها.

ماهي المراحل التي يمر بها البرنامج بلغة C++ قبل تنفيذه؟

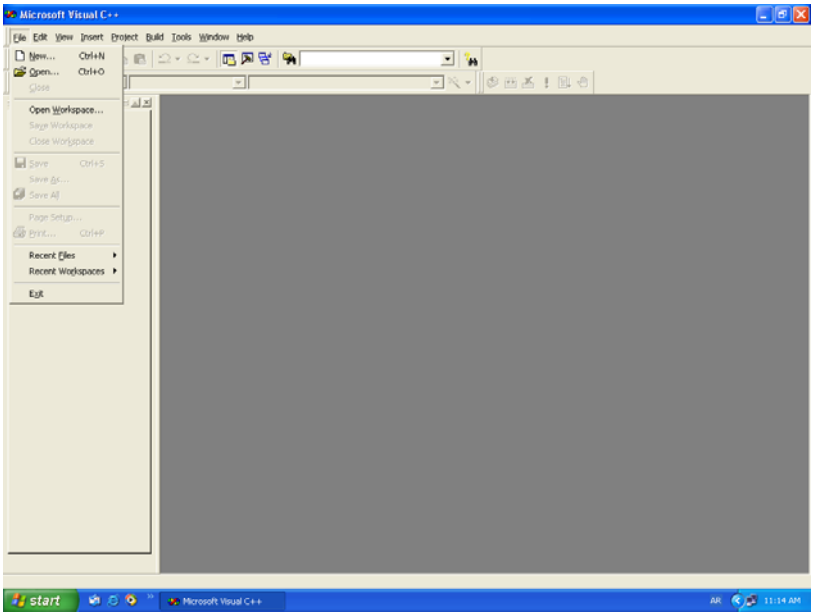
يمر البرنامج بلغة سي ++ بست مراحل قبل تنفيذه وهي:

1. كتابة البرنامج. Edit
2. مرحلة ما قبل الترجمة او المعالجة Preprocess
3. الترجمة Compile
4. الربط Link
5. التحميل Load
6. التنفيذ Execute

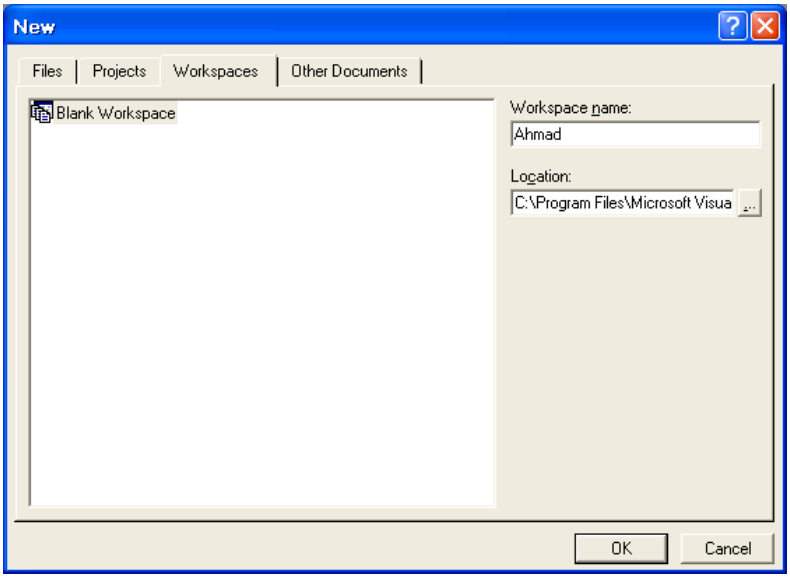
في هذا القسم – عزيزي الطالب – سوف نشرح كيفية كتابة أول برنامج لك بلغة الـ C++ وباستخدام Microsoft Visual C++ وهي بيئة تطويرية أخرى يمكن أن تتعامل معها من أجل التمكن من الانتقال إلى برمجة تطبيقات ويندوز باستخدامها.

- من أجل البدء بأي مشروع قم بالخطوات التالية:

1- اختر New من القائمة File



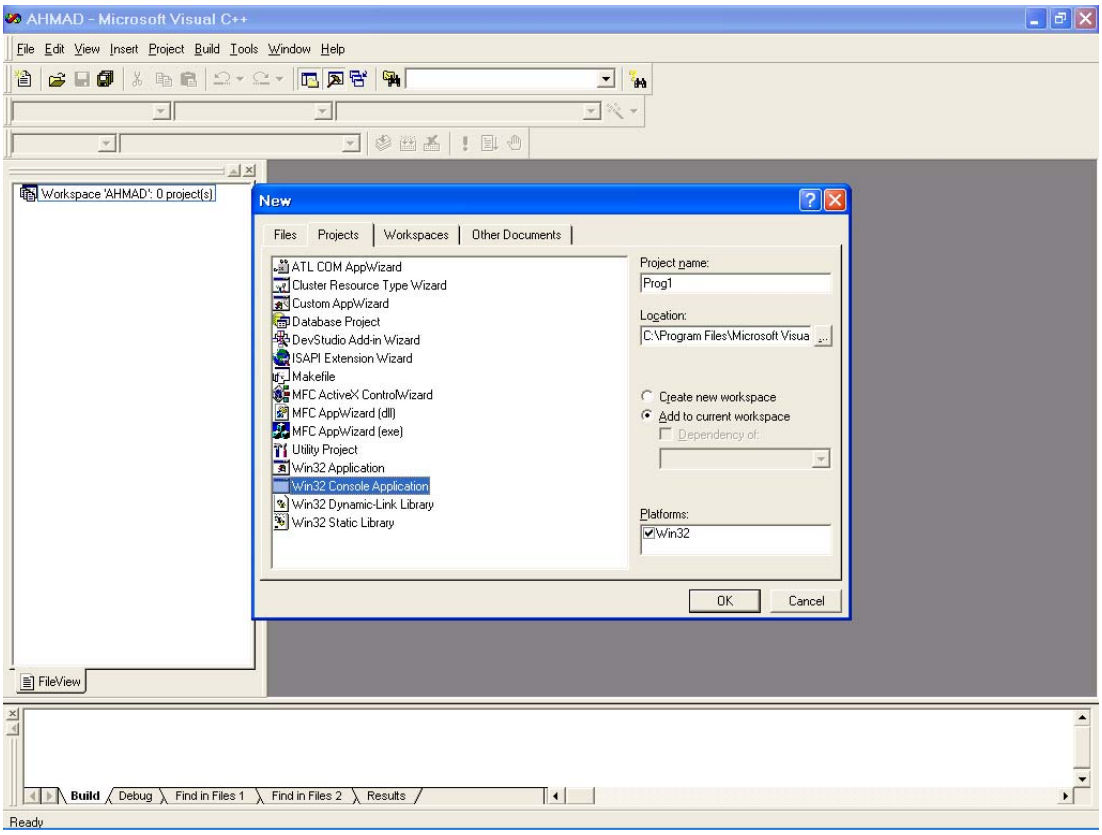
2- تحصل على النافذة كما في الشكل التالي، من الصفحة Workspaces سيظهر Blank Workspace وهو مكان العمل الذي سوف توضع فيه مشاريعك.



3- عند خانة Workspace name أدخل اسم (مثلاً اسمك)، يمكنك تغيير مكان التخزين عن طريق الخانة Location.

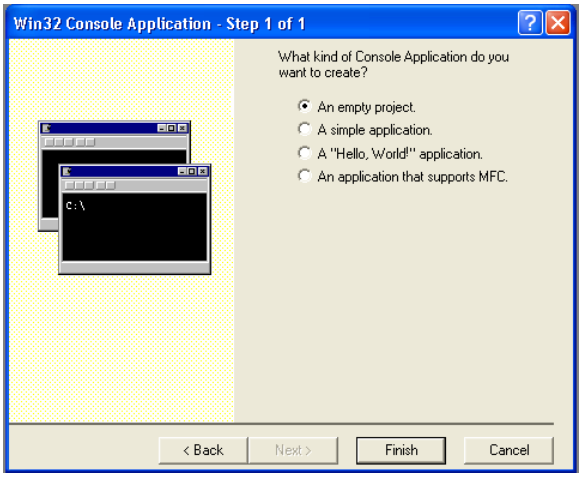
4- الآن ولنقوم بتشكيل أول مشروع يجب علينا اختيار الأمر New من القائمة File لتحصل على نفس الشاشة التي حصلت عليها مسبقاً ولكن الصفحة Projects هي التي ستظهر أمامك بدلاً عن Workspaces.

5- اختر Win32 Console Application، على يمين النافذة وعند خانة Project Name اكتب اسم مشروعك وليكن Prog1، بشكل افتراضي سوف يقوم الفيجول سي++ بإنشاء دليل بنفس اسم المشروع ويضع فيه جميع الملفات التي سوف تضيفها لاحقاً لمشروعك.



6- يجب اختيار Add to current workspace ليتم إضافة مشروعك Prog1 إلى الـ Workspace الذي قمت بإنشائه في الخطوتين 2-3.

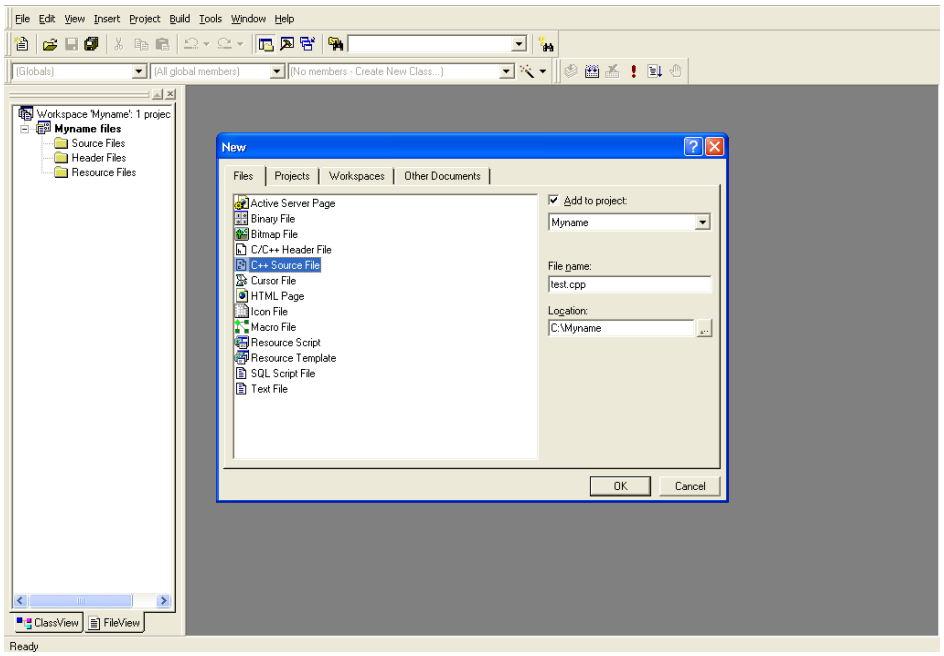
7- اضغط على زر OK لتحصل على النافذة التالية



8- تأكد من اختيار An empty Project ومن ثم اضغط على زر Finish.

9- الآن ولنقوم بكتابة أول برنامج يحب علينا إنشاء ملف جديد باختيار الأمر New من القائمة File لتحصل على نفس الشاشة التي حصلت عليها مسبقاً ولكن الصفحة Files هي التي ستظهر أمامك بدلاً عن Projects.

10-اختر C++ Source file واكتب اسم ملفك عند خانة File Name وليكن test واضغط زر Ok.



11-لاحظ أن في مجلد Source files تم إضافة اسم الملف test.cpp.

12-ابدأ بكتابة البرنامج التالي:

```
#include <iostream.h>
void main()
{
int x;
cout<<"Please enter a number: ";
cin>>x;
cout<<"The number is: ";
cout<<x<<endl;
}
```

يجب الانتباه إلى حالة الأحرف الكبيرة والصغيرة لأن لغة الـ C++ تفرق بينهما.

13-لتنفيذ البرنامج السابق وبناء ملف تنفيذي قابل للتنفيذ بأي وقت قم بالخطوات التالية:

أ- من القائمة Build اختر الأمر compile test.cpp سوف يتم ترجمة الملف وتحويله من اللغة المقروءة إلى لغة الآلة وذلك بعد فحصه من الأخطاء وتصحيحها.

ب- من القائمة Build اختر الأمر Build ليتم تشكيل الملف التنفيذي test.exe.

ت- من القائمة Build اختر الأمر Execute test.exe لتنفيذه.

ث- إذا كان كل شيء صحيح سوف تظهر نافذة مثل نافذة نظام التشغيل دوس في أعلاها اسم الملف test.exe

ج- ظهرت لك عبارة Please enter a number والآن هو في حالة انتظار إدخال العدد .

ح- أدخل العدد واضغط Enter↵.

خ- سوف تجد العبارة The number is والعدد الذي قمت بإدخاله.

- **عليك الآن إكمال كتابة البرنامج التالي ليقوم بإضافة العدد 2 إلى العدد المدخل وطباعة الناتج.**

- استخدم الخطوات 4 حتى 10 لإضافة مشروع جديد إلى نفس الـ Workspace وليكن اسمه prog2.
- قم بكتابة البرنامج بعد أن تكمل الفراغات:

```
#include <iostream.h>
void main()
{
const int s=2 ;
int  n , sum ;
cout<<" Enter a number: " ;
cin>>n ;
sum = n + s ;
cout<<"plus 2= "<<sum<<endl;
}
```

- انتقل إلى مرحلة تنفيذ البرنامج (انتبه أنه سيقوم بتنفيذ المشروع الثاني وذلك بضغط زر اليمين على prog2 وتفعيل هذا المشروع باختيار Set as Active project).

تدريب: بالاستفادة من المعلومات التي قدمت لك في القسمين السابقين، حاول – عزيزي الدارس– أن تنشيء مشروعا جديد prog3 ضمن أحد بيئات التطوير السابقة ليقوم بحساب مساحة دائرة حسب النصف قطر المدخل وطباعة الناتج.