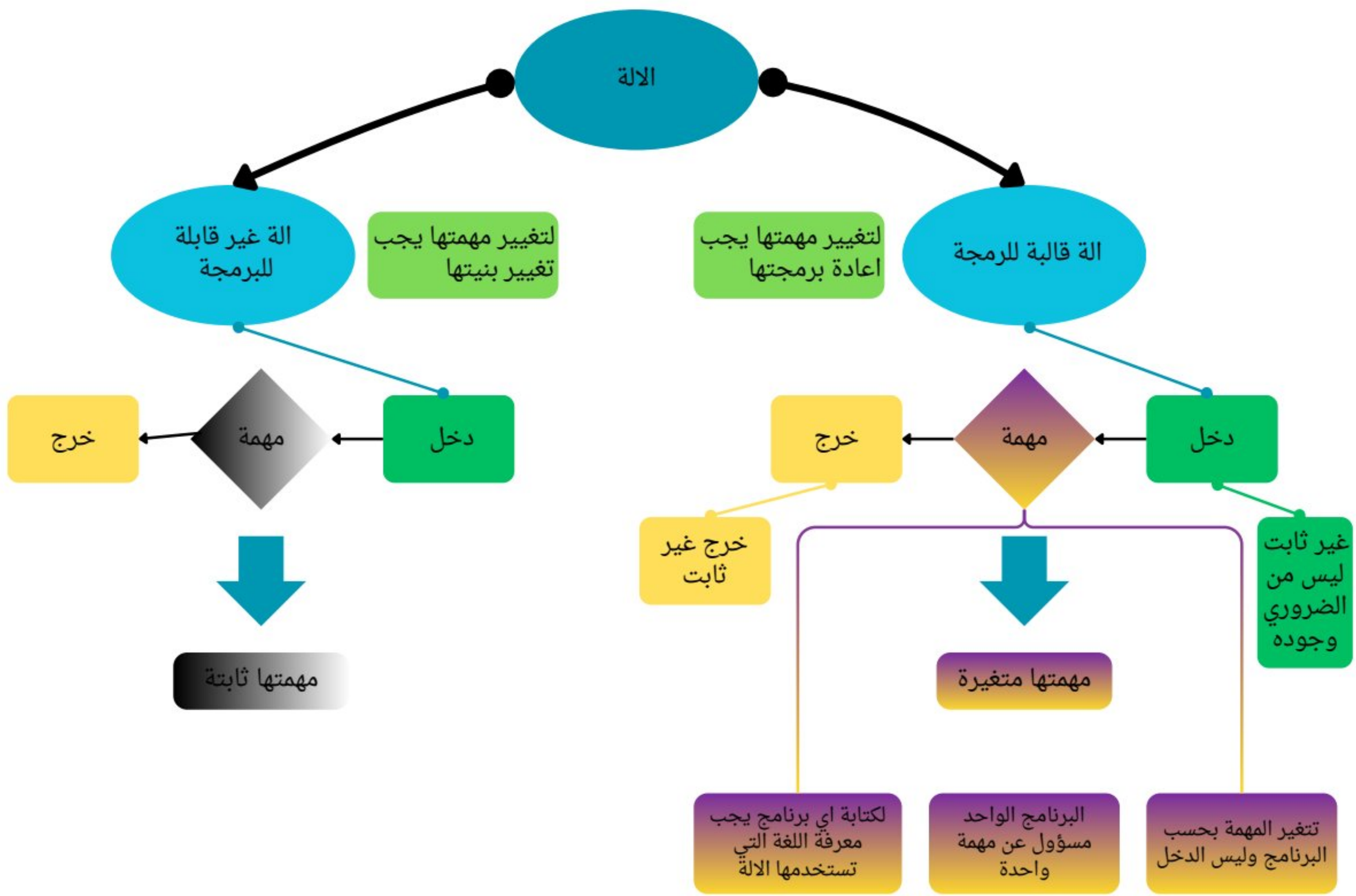
# ملخص مبادى برمجة

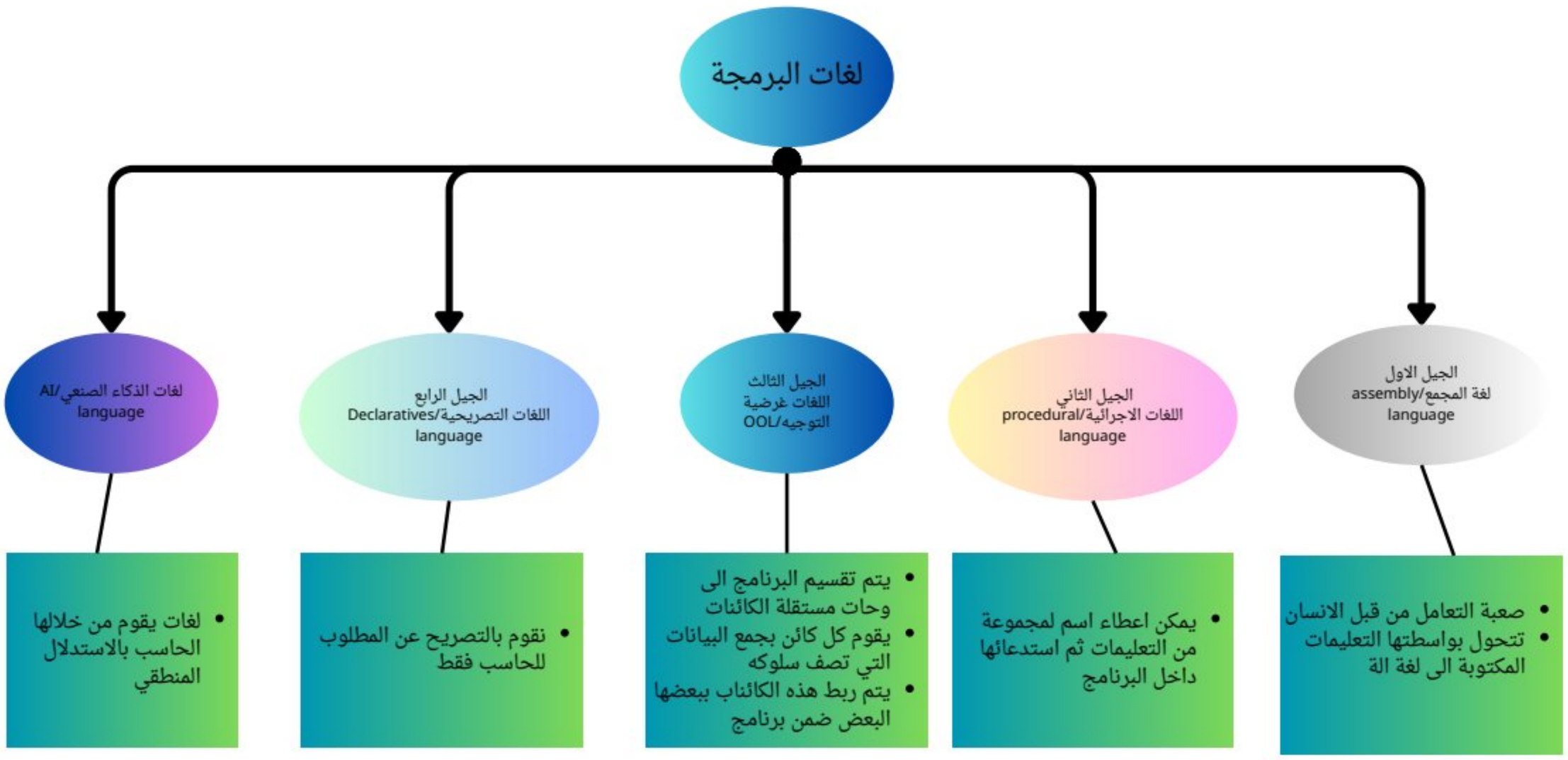
## المحاضرة الاولى

## الالات




الممسوحة ضوئيا بـ CamScanner

# لغات البرمجة

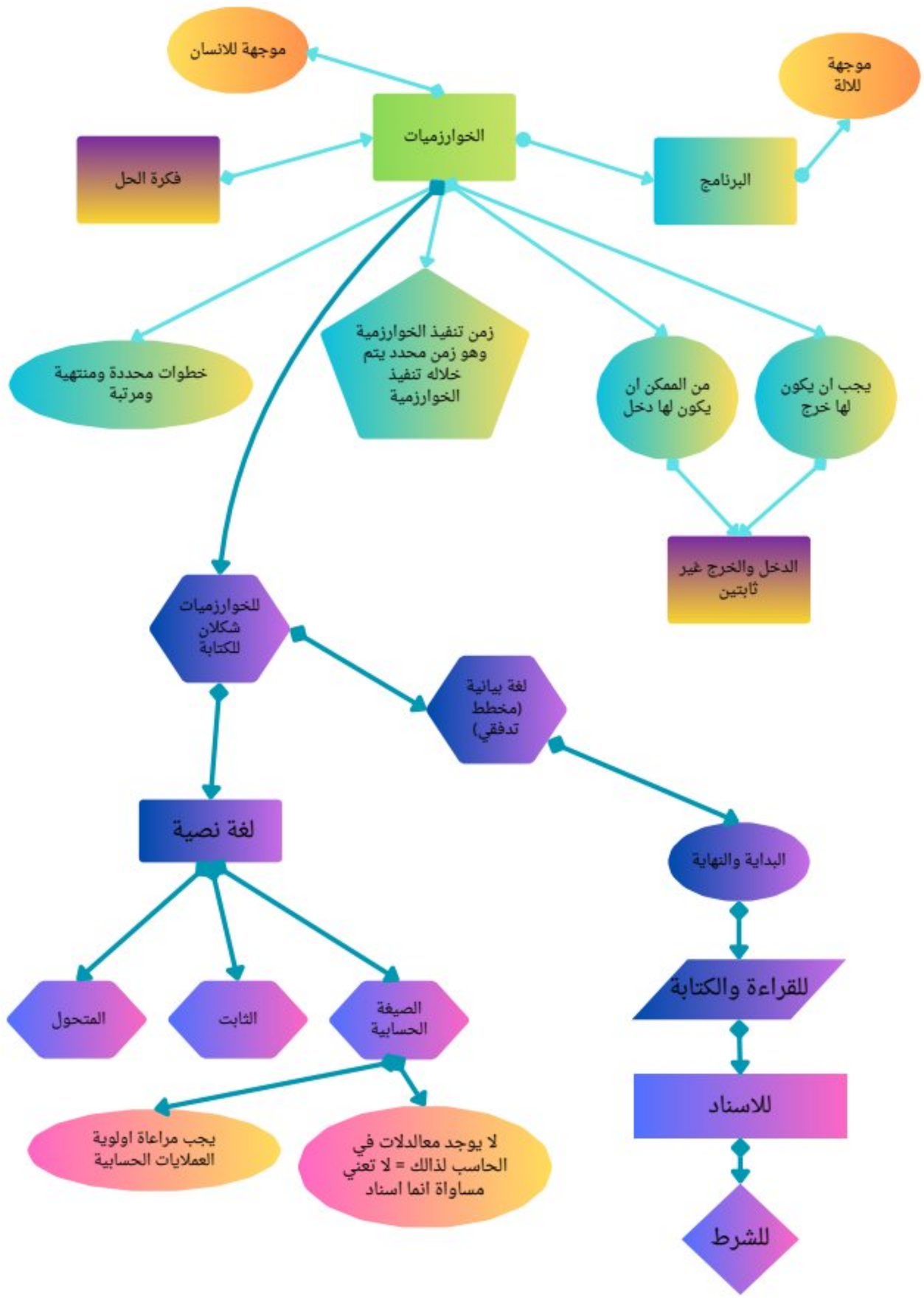



الممسوحة ضوئيا بـ CamScanner

# الخوارزميات

:تعريف الخوارزميات

هي مجموعة من الخطوات الرياضية والمنطقية والمتسلسلة ← لحل **مسألة** ما او القيام **بمهمة** محدد

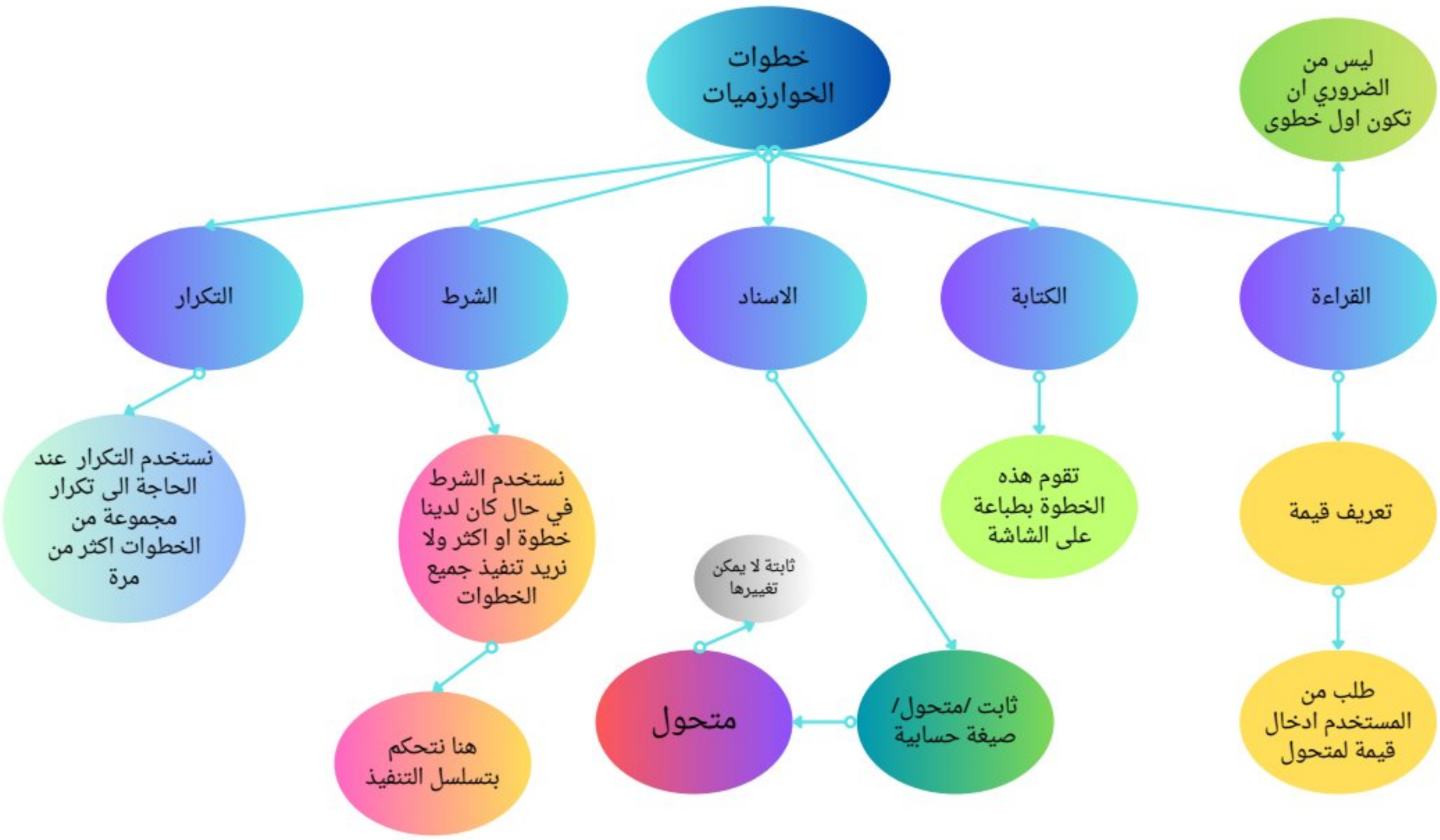



**المتحول**: هو متغير يخزن فيه قيمة لحظية و تبقى هذه القيمة ثابتة حتى يتم تغييرها من قبل **الخوارزمية**

الممسوحة ضوئيا بـ CamScanner

# المحاضرة الثالثة

# خطوات الخوارزميات




الممسوحة ضوئيا بـ CamScanner

# ملاحظات هامة

## القراءة

### أقرء x :

1- عرف قيمة x

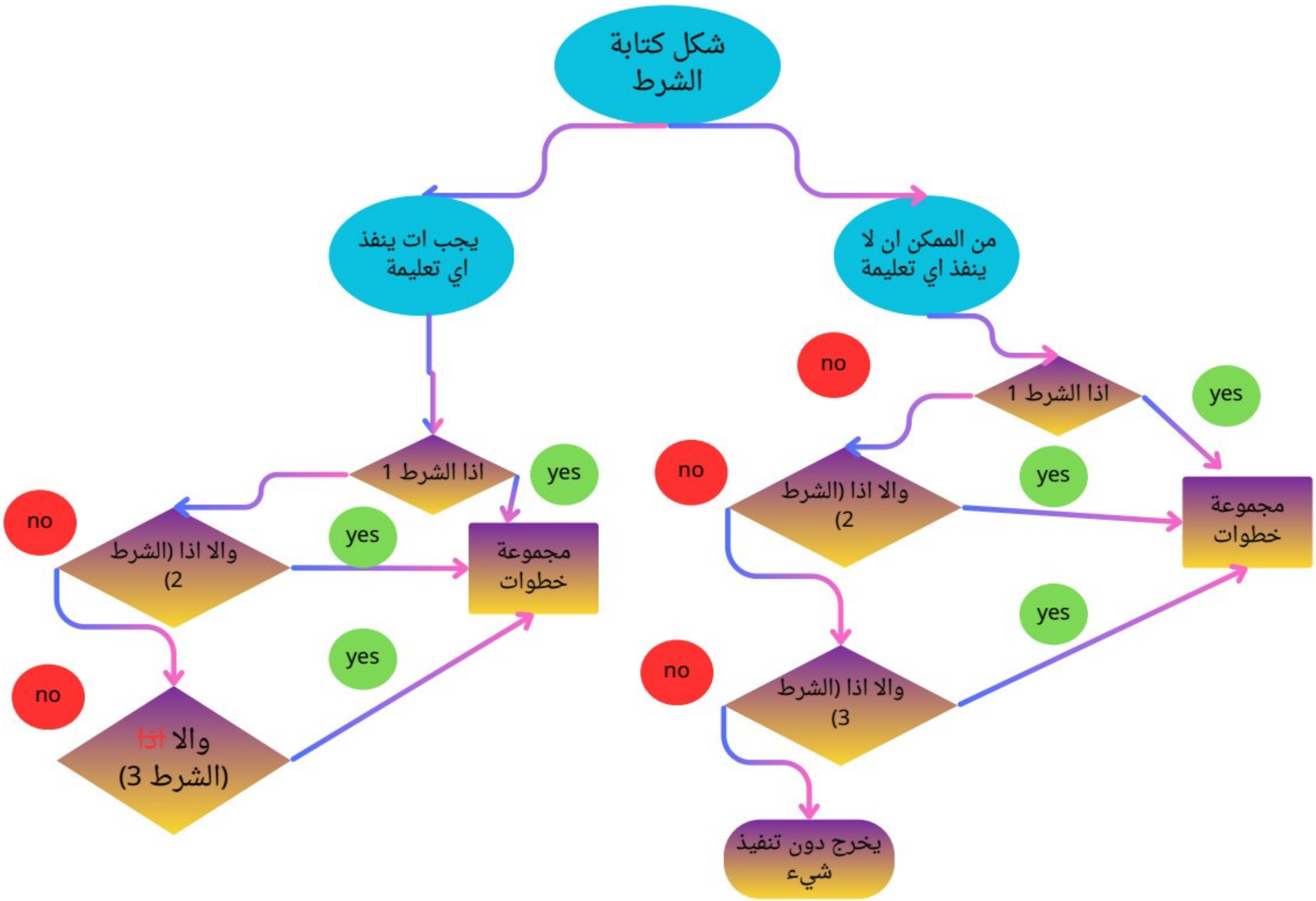
2- اطلب من المستخدم ادخال قيمة x

## الكتابة

| المثال | نوعه | الخرج |
|---|---|---|
| اكتب x | متحول | قيمة x |
| اكتب 2 | ثابت | العدد 2 |
| اكتب "hello" | ثابت | سلسلة المحارف hello |
| اكتب x | ثابت | المحرف x |
| اكتب x+y | صيغة حسابية | قيمة المجموع x+y |

الممسوحة ضوئيا بـ CamScanner

# الية عمل الشرط




ملاحظة: يجب ان يكون الشرط بصيغة مقارنة والنتائج اما محققة او غير محققة

## الفرق بين القرائة والاسناد

| | |
|---|---|
| القرائة | يقوم المبرمج بادخال قيمة المتحول |
| الاسناد | يقوم المبرمج بتعديل قيمة المتحول |

الممسوحة ضوئيا بـ CamScanner

# بعض الاسألة المهمة التي يجب معرفة حلها

## اكتب خوارزمية لجمع عددين

| |
|---|
| اقرأ x |
| اقرأ y |
| اكتب z= x+y |

## اكتب خوارزمية لتمييز عدد موجب او سالب

| |
|---|
| اقرأ العدد |
| اقرأ (عدد⟵x ) |
| إذا (x > 0) |
| اكتب «موجب» |
| و إلا إذا(x < 0) |
| اكتب «سالب» |
| و إلا |
| اكتب «صفر» |

## ألية كتابة الشرط

**الشكل العام لكتابة الشرط:**

```
إذا (الشرط ١)
    مجموعة خطوات
و إلا إذا (الشرط ٢)
    مجموعة خطوات
و إلا
    مجموعة خطوات
```

```
إذا (الشرط ١)
    مجموعة خطوات
و إلا
    إذا (الشرط ٢)
        مجموعة خطوات
    و إلا
        إذا (الشرط ٣)
            مجموعة خطوات
```

## لمعرفة عدد خانات عدد موجب يتم ادخاله من قبل المستخدم

لحل رقم (١):

| |
|---|
| اقرأ العدد |
| اقرأ (عدد⟵x ) |
| إذا (x<10 و x ≥ 0) |
| اكتب «خانة واحدة » |
| إذا (x<100 و x ≥ 10) |
| اكتب «خانتان» |
| إذا (x<1000 و x ≥100) |
| اكتب «ثلاث خانات» |
| إذا (x ≥1000) |
| اكتب «أربع خانات» |

الحل رقم (٢):

| |
|---|
| اقرأ العدد |
| اقرأ (عدد⟵x ) |
| إذا (x<10 و x ≥ 0) |
| اكتب «خانة واحدة » |
| و إلا إذا (x<100) |
| اكتب «خانتان» |
| و إلا إذا (x<1000) |
| اكتب «ثلاث خانات» |
| و إلا |
| اكتب «أربع خانات» |

الممسوحة ضوئيا بـ CamScanner

# لطباعة الاعداد من 1-100

الحل رقم (١)

| |
|---|
| x ← 0 |
| إذا (x<100) |
| x ← x+1 |
| اكتب x |

الحل رقم (٢)

| |
|---|
| x ← 1 |
| إذا (x≤100) |
| اكتب x |
| x ← x+1 |

الحل رقم (٣)

| |
|---|
| x ← 1 |
| إذا (x≤100) |
| x ← x+1 |
| اكتب x-1 |

الممسوحة ضوئيا بـ CamScanner

# المحاضرة الرابعة

## C++

```
# include <iostream>
using namespace std;
int main()
{
// مجموعة التعليمات
return 0;
}
```

يتضمن

تعني ان هذا السطر موجه للمترجم

اسم مكتبة

قياسي : بتعامل مع المكاتب بشكل قياسي كأولوية

تكتب من اجل ان يعرف البرنامج اي مكتبة لها الالولوية

نفذ: احجز مكان في الذاكرة

interger:عدد صحيح

تابع رأيسي لعدد صحيح

طبعا ليس بالضرورة الرقم 0 نكتب هاذا السطر من اجل التأكد من ان الكود لا يعاني من اي bug و يكتب في اخر النتيجة 0

السطرة الاول: نطلب من المترجم استدعاء مكتبة <iostream>

<iostream> هي مكتبة تحتوي لغة الالة وهي مسؤلة عن عملات الادخال والاخراج

السطرة الثاني: إن <<namespace >> هي أداة عابرة للمكتبات تمنع تعارضها مع بعضها البعض و تدل على المكتبة المقصودة

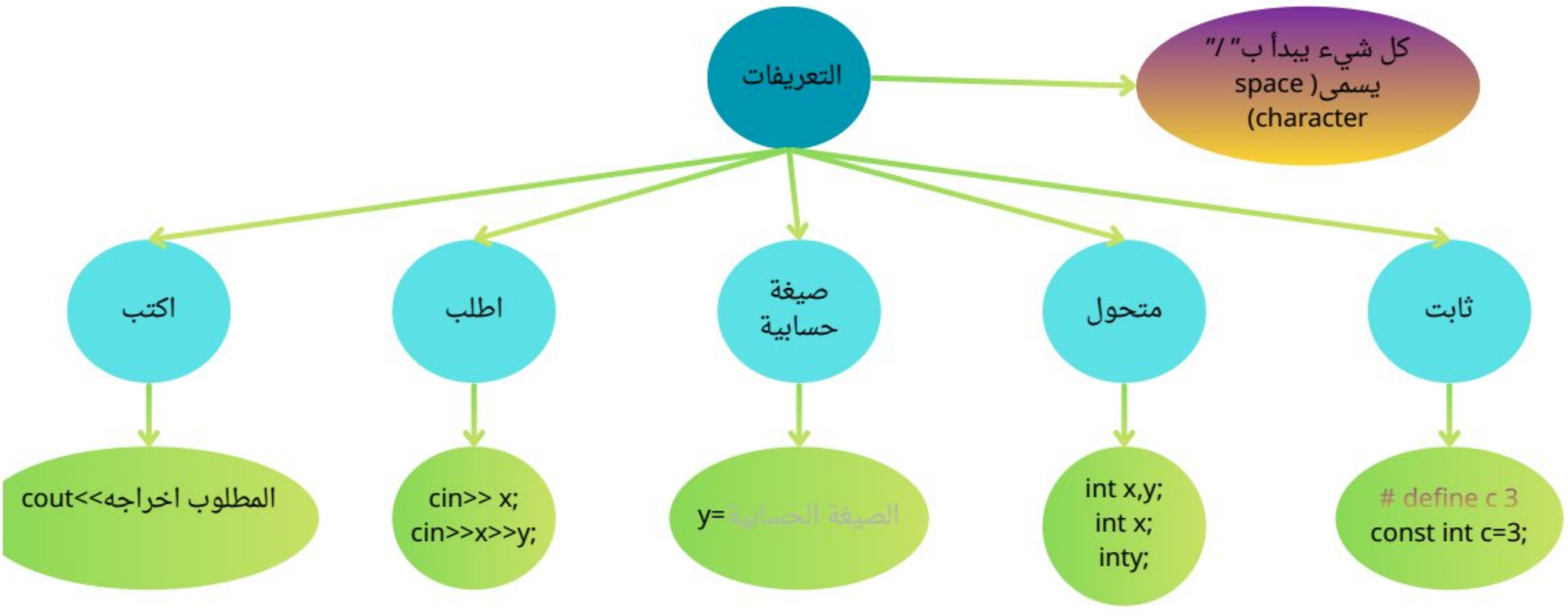
السطر الثالث: إن <<main>> هو التابع الرئيسي للبرنامج ويكون تنفيذ البرنامج محصور بالتعليمات ضمن أقواس المجموعة التي تلي التابع

الممسوحة ضوئيا بـ CamScanner

# لغة C++ حساسة ل:

| لغة C++ | حساسة | غير حساسة |
|---|---|---|
| الاحرف الكبيرة | ✔ | ✘ |
| المسافات | ✘ | ✔ |
| (enter) | ✘ | ✔ |

# كيفية التعريف في لغة c++



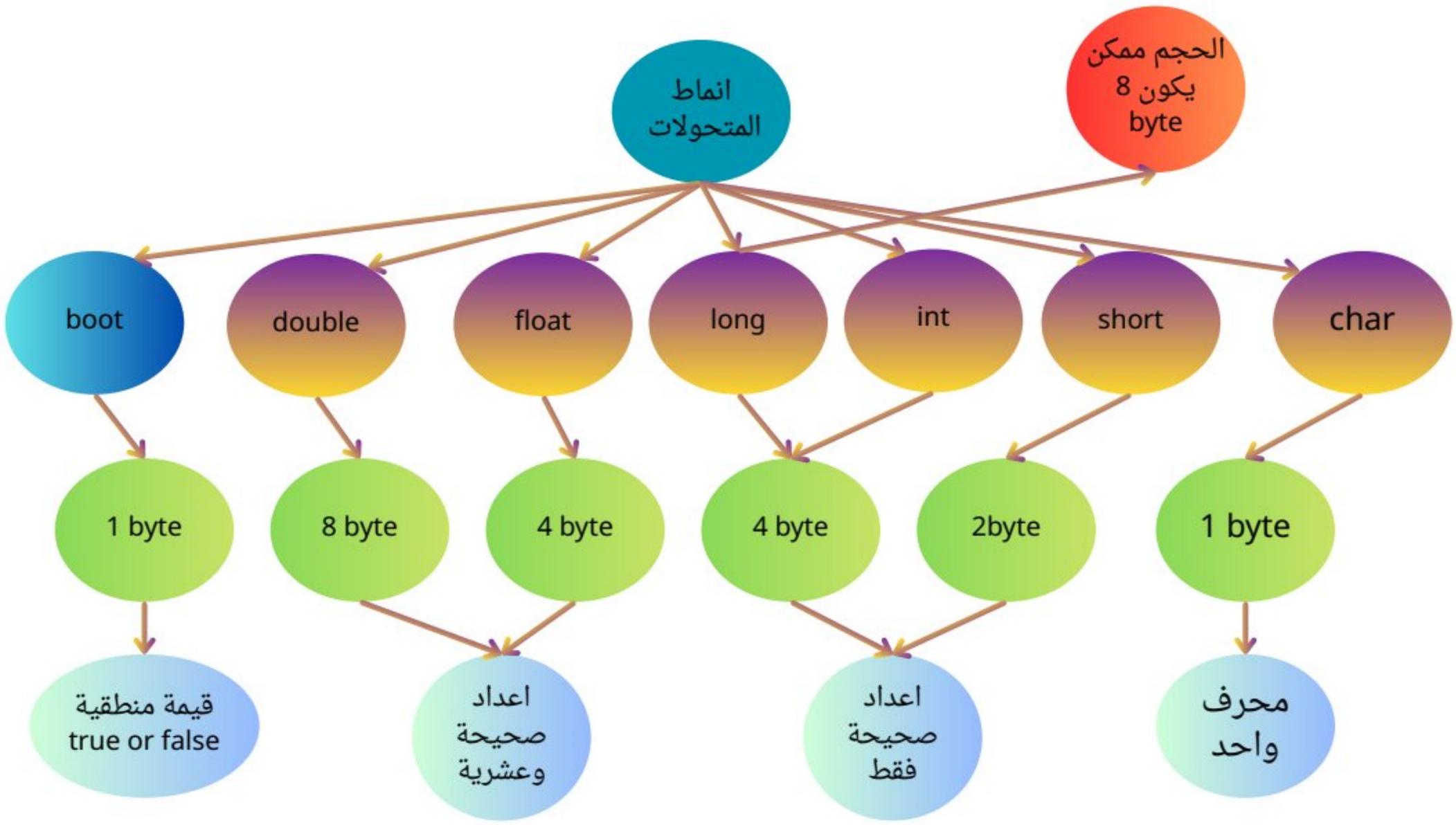

انتبه: من الخطأ كتابة ← `cin >> x,y;`

لو كتبنا الكود
**int x;**
**cout<<x;**

لان كل متحول له قيمة
ولا يوجد متحول بلا قيمة

- سيطبع البرنامج قيمة عشوائية
- تكون القيمة مأخوذة من مكان تخزين المتحول

الممسوحة ضوئيا بـ CamScanner



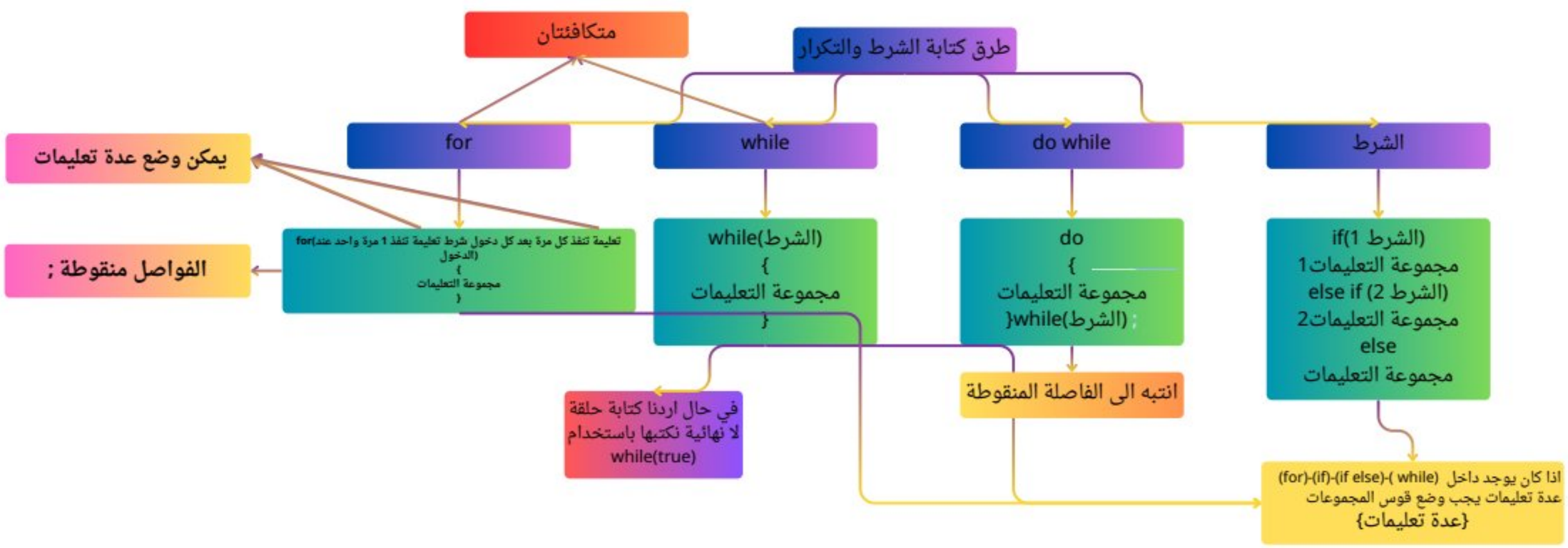

## بعض الكلمات المفتاحية التي تعطي معنى اضافي لنوع البيانات

| | Modifier | Tpe | Name; |
|---|---|---|---|
| 8 byte | Long | Int | name1; |
| 2 byte | Short | Int | name2; |
| قيم سالبة وموجبة | Signed | Int | name3; |
| قيم موجبة فقط | unsigned | int | name4; |

الممسوحة ضوئيا بـ CamScanner

# المحاضرة الخامسة

# كتابة الشرط و التكرار




| while | do while |
|---|---|
| 1. يقوم البرنامج بفحص الشرط<br>2. ينفذ التعليمات ويعود لفحص الشرط | 1. يقوم البرنامج بتنفيذ التعليمات<br>2. يفحص الشرط<br>3. ويعيد تنفيذ التعليمات بناء على النتيجة |
| من الممكن ان لا يدخل البرنامج لتنفيذ التعليمات | يدخل على الاقل مرة واحجة |

## لكتابة برنامج من يطبه الارقام من 1 الى n باستخدام while

الممسوحة ضوئيا بـ CamScanner

```
# include <iostream>
using namespace std;
int main()
{
int n,x=1;
cin>>n;
while (x<=n)
{cout<<x;
x++;}
return 0;
}
```

النتيجة

```
5
1
2
3
4
5

Process returned 0 (0x0)   execution time : 3.681 s
Press any key to continue.
```

يمكن اختصاره بسطر
cout<<x++<<endl;

## برنامج لمعرفة نوع عدد مدخل

```
# include <iostream>
using namespace std;
main()
{
    int x;
    cin>>x;
    if(x>=0)
        cout<<"positive";
    else
        cout<<"negative";
    return 0;
}
```

النتيجة

```
25
positive
Process returned 0 (0x0)   execution time : 3.548 s
Press any key to continue.
```

## ملاحظة:عند كتابة شرط التساوي يجب كتباته مثلا while(x==0)

الفرق بين x++و++x

```
# include <iostream>
using namespace std;
main()
{
    int x=2;
    cout<<x++;
    return 0;
}
```

```
2
Process returned 0 (0x0)   execution time : 0.076
Press any key to continue.
```

```
# include <iostream>
using namespace std;
main()
{
    int x=2;
    cout<<++x;
    return 0;
}
```

```
3
Process returned 0 (0x0)   execution time :
Press any key to continue.
```

## برنامج لطباعة الاعداد من 1 الى n باستخدام do while

الممسوحة ضوئيا بـ CamScanner

```
# include <iostream>
using namespace std;
main()
{

    int n,x=1;
    cin>>n;

    do{
        cout<<x<<endl;
        x=x+1;
    }while(x<=n);
    return 0;
}
```

النتيجة

```
5
1
2
3
4
5

Process returned 0 (0x0)   execution time : 2.109 s
Press any key to continue.
```

## اكتب برنامج لطباعة الاعداد من 1 الى n باستخدام for

```
# include <iostream>
using namespace std;
main()
{

    int n,x=1;
    cin>>n;

   for(x=0;x<=n;x++)
   {
       cout<<x<<endl;
   }
    return 0;
}
```

النتيجة

```
5
0
1
2
3
4
5

Process returned 0 (0x0)   execution time : 2.590 s
Press any key to continue.
```

| طريقة كتابة break | طريقة كتابة continue |
|---|---|
| • تستخدم لكسر الحلقة<br>• لما بوصل البرنامج لعند break بوقف وبيطلع برا الحلقة | • لما بوصل البرنامج لعند هاي الحلقة بوقف وبيرجع بيفحص الشرط اذا تحقق بنفذ التعليمات |

## مثال عن continue و break

الممسوحة ضوئيا بـ CamScanner

```
# include <iostream>
using namespace std;
main()
{

    int x=1;
    while(x<=100)
    {
        if(x==5)
            break;
        cout<<x<<endl;
        x++;
    }
    return 0;
}
```

```
1
2
3
4

Process returned 0 (0x0)   execution time : 0.085 s
Press any key to continue.
```

```
# include <iostream>
using namespace std;
main()
{

    int x=1;
    while(x<=100)
    {
        if(x==5)
            continue;
        cout<<x<<endl;
        x++;
    }
    return 0;
}
```

```
1
2
3
4
```

البرنامج لسا ما انتهى
علقان جوا الحلقة

الممسوحة ضوئيا بـ CamScanner

# المحاضرة السادسة

## المصفوفات

- **التعريف**
  - هي مجموعة من البيانات التي تشبه بعضها
  - لها طول وحجم ثابتين
    - ولا يمكن تخزين مصفوفة فوق اخرى
- **انواع المصفوفات**
  - المصفوفات الاحادية
    - int x[n];
      - هو عدد العناصر n
        - عند تعريف اي مصفوفة تحجز مكان في الذاكرة
          - int x[n]
          - هنا تحجز nx4 byte في الذاكر
            - هاكذا يتم الوصول الى العناصر في الذاكرة
          - طبعا نمط المصفوفة لها دور في كمية الحجز
          - حجز العناصر يكون بشكل متتالي
          - يمكن مناداة عنصر من خلال عنصر أخر x[1+x[3]];
      - تبدا العناصر من الصفر x[0]=a;
    - يمكن تعريف المصفوفة بعدة طرق
      - const int n=10; int x[n;]
      - int x[]={5,6,10}
      - int n; cin >> n; int a [n];
    - ماذا لو كتبنا في ابرنامج
      - int a[100]=10;
        - هنا معظم لغات العمل تعطي error لكن في c++سيقوم البرنامج بوضع 10 في اول عنصر ويملا الباقي تلقائيا
  - المصفوفات الثنائية
    - int x[i][j];
    - لتعريف نقوم باستعمال حلقتين
      - ادخال بالاسطر

```
for (int j=0;j<1000;j++)
  for(int i=0;i<6;i++)
    cin>>marks[j][i];
```

      - ادخال بالاعمدة

```
for (int j=0;j<1000;j++)
  for(int i=0;i<6;i++)
    cin>>marks[i][j];
```

Presented with xmind

الممسوحة ضوئيا بـ CamScanner

# المحاضرة السابعة

# التوابع

هي مجموعة من التعليمات لها اسم معين

1. يتم استخدام التابع لجعل الكود اكثر سهولة في القراءة
2. لا يمكن استدعاء تابع دون تعريفه

# الفرق بين التعريف والاستدعاء

```
# include <iostream>
using namespace std;
    int add(int x,int y)
    {return (x+y);}

main()
{
int a,b,z;
cin>>a>>b;
z=add(a,b);
cout<<z;
}
```

ليس بالضروري ان يكون main هو اول تابع مثل هنا

هون تم تعريف التابع

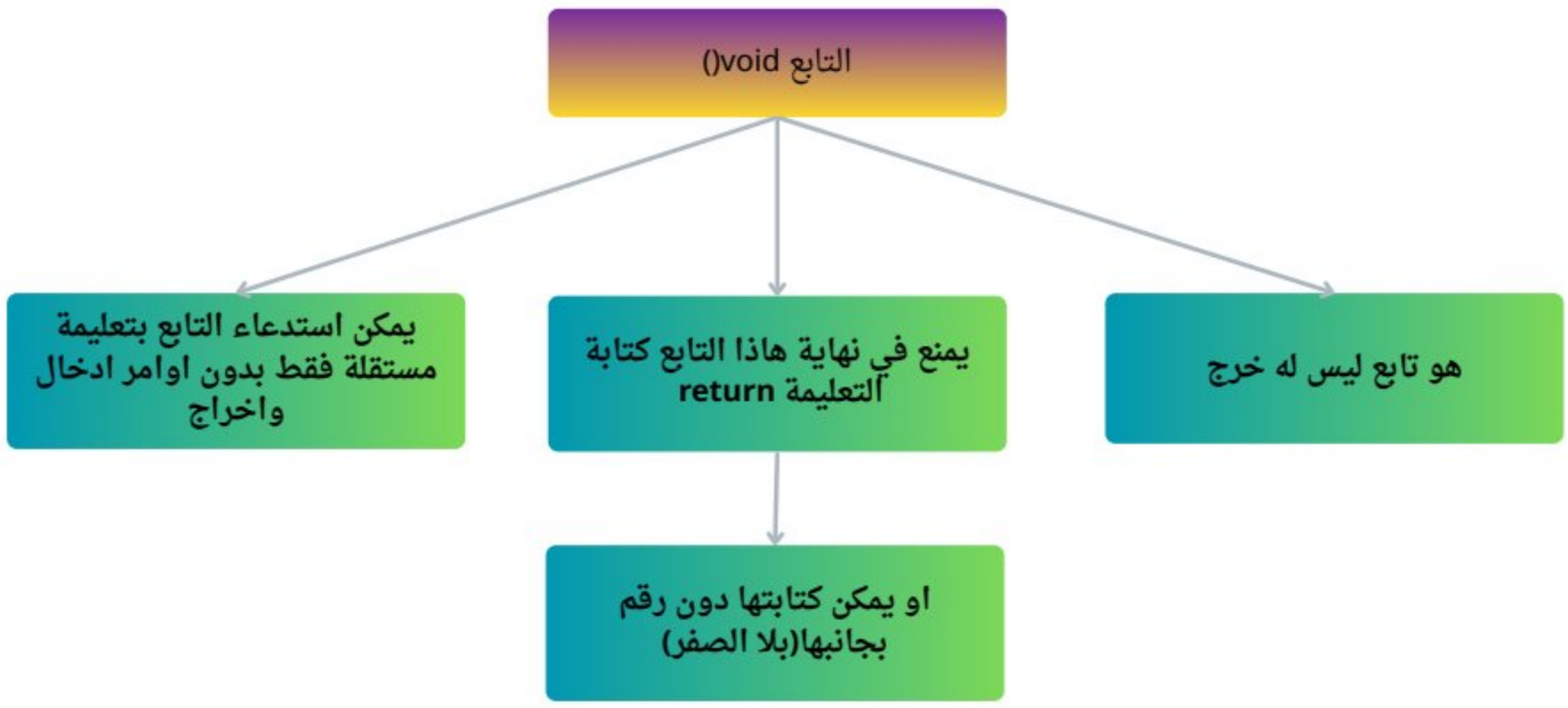
وهون تم استدعائه

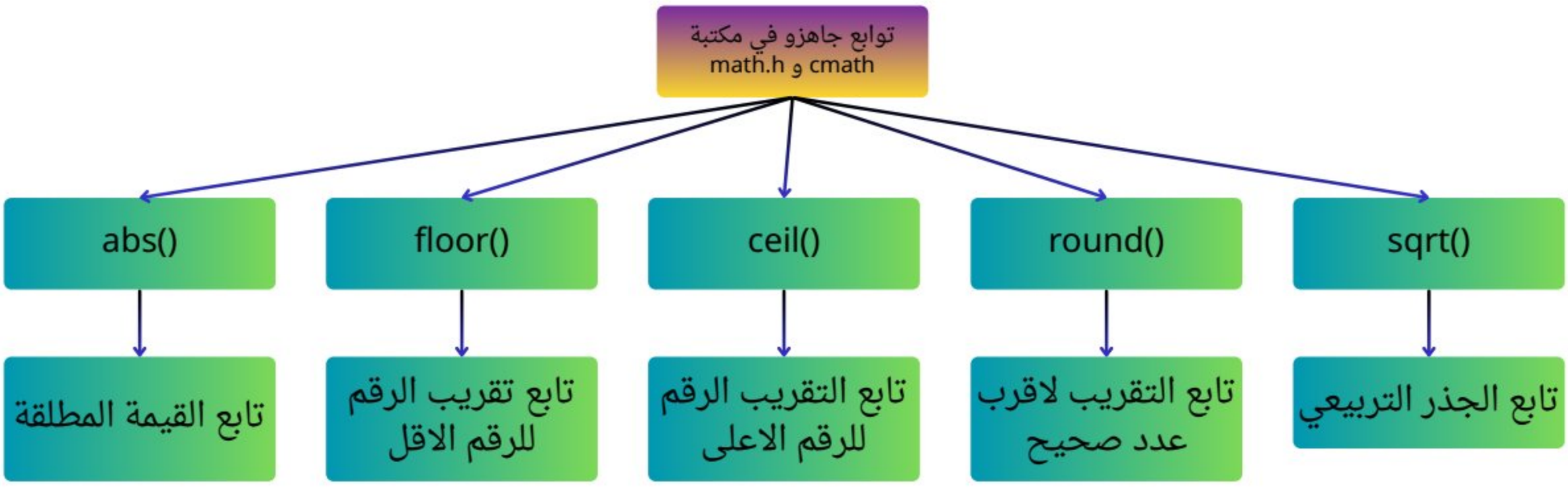
يمكن استعاء التابع عن طريق

- قراءة
- كتابة
- مساواة

الممسوحة ضوئيا بـ CamScanner

# التابع void()




الممسوحة ضوئيا بـ CamScanner

# المحاضرة الثامنة

# التوابع

```
# include <iostream>
using namespace std;
    int add(int x,int y)
    {return (x+y);}

main()
{
int a,b,z;
cin>>a>>b;
z=add(a,b);
cout<<z;
}
```

هون تم تعريف التابع

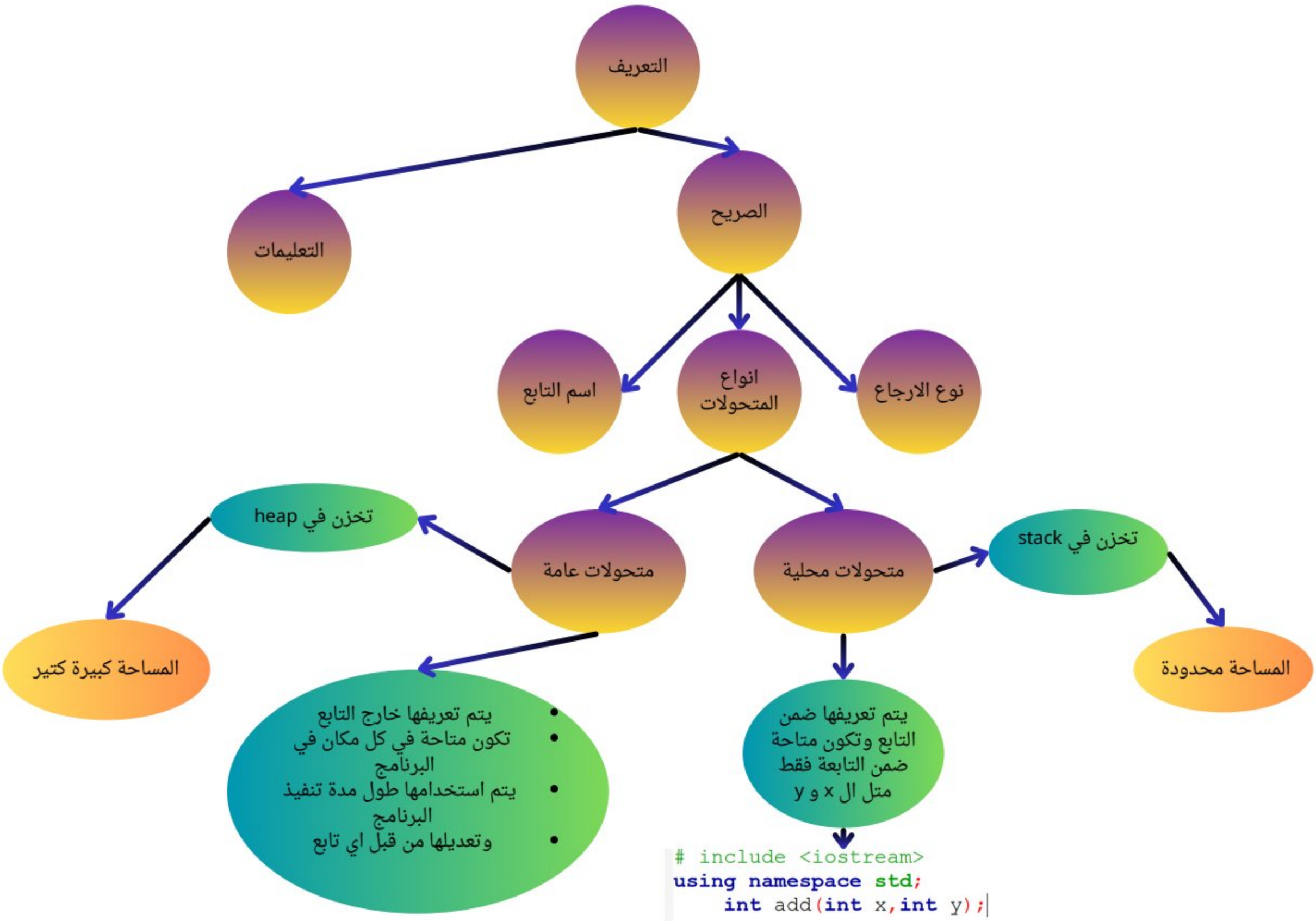
هون تم التصريح عنو فقط
الفرق انو مافي مجموعة تعليمات

```
# include <iostream>
using namespace std;
    int add(int x,int y);

main()
{
int a,b,z;
cin>>a>>b;
z=add(a,b);
cout<<z;
}
```

## الفرق الجوهري بين التصريح والتعريف وانواع المتحولات

الممسوحة ضوئيا بـ CamScanner




```
# include <iostream>
using namespace std;
    int add(int x,int y);

main()
{
int a,b,z;
cin>>a>>b;
z=add(a,b);
cout<<z;
}
```

التحميل الزائد: باختصار هو لما يكون عندك نفس الاسم والنوع تبع التابع بس في زيادة او نقص متحول او ايكون المتحول غير نوع
مثال:

```
int add (int a, int b)
{
  return a+b;
}

int add (int a, int b, int c)
{
  return a+b+c;
}
```

الممسوحة ضوئيا بـ CamScanner

ملاحظة مهمة عندما تعرف تابع بداخله مصفوفة مافي داعي تكتب عدد عناصر المصفوفة متل هون

```
void print (int arr[])
```

باقي كيفية كتابة تابع يقوم بطباعة جميع العناصر
كيفية كتباة تابع يقوم بتطباعة عناصر المصفوفة الثانية
ما ذكرتهن هون

الممسوحة ضوئيا بـ CamScanner