

مقرر برمجة 1
جلسة العملي الأولى والثانية

الفصل الأول 2025-2026

التوابع Functions

ما هو التابع Function؟

التابع هو برنامج جزئي يقوم بتنفيذ مهمة محددة ويمكن استدعاؤه في أي مكان في البرنامج الرئيسي.

يهدف استخدام التوابع عند كتابة البرنامج لتحقيق عدة ميزات مهمة:

- ❖ إعادة الاستخدام: تستخدم التوابع لتحقيق إعادة الاستخدام للكود ذاته في أماكن مختلفة في البرنامج من خلال استدعاء التابع عدى مرات في البرنامج عند الحاجة إليه.
- ❖ تجزئة الكود البرمجي: يفيد تعريف التابع في تجزئة الكود البرمجي إلى وحدات صغيرة تسهل إدارتها وفهمها والتعامل معها.
- ❖ التقليل من تكرار التعليمات البرمجية: يسهم استخدام التوابع في تقليل كمية التعليمات البرمجية المتكرر استخدامها في أماكن مختلفة من البرنامج.

الشكل العام للتابع:

```
(Data type returned) function name (parameter1, parameter 2, ..) {
```

```
Instruction 1
```

```
Instruction 2
```

```
..
```

```
Return value // if not the returned value is void, otherwise we don't return any value.
```

```
}
```

مثال (1):

تابع لا يأخذ معاملات ولا يعيد أي قيمة

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

void greet() {
    cout << "Hello Students!" << endl;
}

int main() {
    greet(); // استدعاء التابع
    return 0;
}
```

مثال (2)

تابع يأخذ معاملات ولا يعيد قيمة

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

void printSum(int a, int b) {
    cout << "Sum = " << a + b << endl;
}

int main() {
    printSum(5, 7);
}
```

مثال (3):

تابع يأخذ معاملات ويعيد قيمة

اكتب برنامج يحسب مربع عدد صحيح:

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

// Return-type function definition
int square(int x) {
    return x * x;
}
```

```
int main() {  
  
    int result = square(5);  
    cout << "Square of 5 is: " << result << endl;  
  
    return 0;  
}
```

مثال (4):

اكتب برنامج يتم فيه إيجاد أكبر عدد بين ثلاثة أعداد مدخلة

```
#include <iostream>  
using namespace std;  
  
// دالة لإيجاد أكبر عدد بين ثلاثة أعداد  
int maxOfThree(int a, int b, int c) {  
    int maxNum = a;  
    if (b > maxNum) maxNum = b;  
    if (c > maxNum) maxNum = c;  
    return maxNum;  
}  
  
int main() {  
    int a, b, c;  
    cout << "ادخل ثلاثة أعداد مفصولة بمسافة " << endl;  
    cin >> a >> b >> c;  
  
    cout << "أكبر عدد هو " << maxOfThree(a, b, c) << endl;  
  
    return 0;  
}
```

تمارين للتدريب:

- (1) اكتب تابع لحساب السرعة ثم استخدم التابع في البرنامج الرئيسي لحساب السرعة تبعاً للمسافة والوقت المدخلين من قبل المستخدم. (ملاحظة للمساعدة على الحل: السرعة = (المسافة / الزمن))
- (2) اكتب برنامج يحسب مجموع الأعداد الزوجية من 1 إلى عدد N مدخل من قبل المستخدم واستخدمه في البرنامج الرئيسي

حل التمارين:

التمرين (1)

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

double speed(double distance, double time) {
    return distance / time;
}

int main() {
    double d, t;
    cout << "Enter distance and time:" :
    cin >> d >> t;

    cout << "Speed = " << speed(d, t) << endl;
    return 0;
}
```

التمرين (2)

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

int sumEven(int n){
    int sum = 0;
    for (int i = 1; i <= n; i++){
```

```
    if (i % 2 == 0)
        sum += i;
    {
    return sum;
    {

int main (){
    int n;
    cout << "Enter n;" :
    cin >> n;

    cout << "Sum of even numbers = " << sumEven(n) << endl;
    return 0;
}
```

نهاية الجلسة