

C# Questions Solutions

حلول أسئلة C# - C# Questions Solutions

دليل المبتدئين - Beginner's Guide

Q1: Define Object Oriented Programming (تعريف البرمجة كائنية التوجه)

الإجابة / Answer:

التي (Objects) هي أسلوب برمجي يعتمد على مفهوم "الكائنات" (OOP) البرمجة كائنية التوجه. تحتوي على بيانات وسلوكيات معاً.

Object-Oriented Programming (OOP) is a programming paradigm based on the concept of "objects" which contain data and behavior together.

المبادئ الأساسية الأربع / Four Main Principles:

1. إخفاء تفاصيل البيانات وإظهار الوظائف الضرورية فقط: **Encapsulation**
 - Hiding data details and showing only necessary functions
2. قدرة كلاس (فئة) على وراثة خصائص وسلوكيات من كلاس آخر: **Inheritance**
 - Ability of a class to inherit properties and behaviors from another class
3. قدرة الكائنات على اتخاذ أشكال متعددة: **تعدد الأشكال (Polymorphism)**
 - Ability of objects to take multiple forms
4. إخفاء التفاصيل المعقدة وإظهار الوظائف الأساسية: **Abstraction**
 - Hiding complex details and showing essential features

مثال بسيط / Simple Example:

```
// クラス 定義 例
public class Car
{
    // 属性 (Properties)
    public string Color;
    public string Model;

    // メソッド (Methods)
    public void Start()
    {
        Console.WriteLine(" - السيارة تعمل" - Car is starting");
    }
}
```

Q2: Write a complete C# program (كتابه) (كامل # برنامح) C#

أعد الـ **array** بـ 10 **integers**، يدخلها **user input** ثم حسبه **odd** **even** **count**
الزوجية والفردية

Required: Program with array of 10 integers, user input, count odd and even numbers

الحل الكامل / Complete Solution:

```
using System;

class Program
{
```

```

static void Main()
{
    // إنشاء مصفوفة تضم 10 أرقام
    // Create an array to store 10 numbers
    int[] numbers = new int[10];

    // متغيرات لـ couting
    // Variables for counting
    int evenCount = 0; // لـ couting الأرقام الزوجية
    int oddCount = 0; // لـ couting الأرقام الفردية

    // طلب من المستخدم إدخال 10 أرقام
    // Ask user to input numbers
    Console.WriteLine("الرجاء إدخال 10 أرقام");
    Console.WriteLine("Please enter 10 integer numbers:");

    for (int i = 0; i < 10; i++)
    {
        Console.Write($"أدخل الرقم {i + 1}: ");
        numbers[i] = int.Parse(Console.ReadLine());
    }

    // استهلاك foreach لـ couting كل عنصر في المصفوفة
    // Using foreach to loop through each element
    foreach (int num in numbers)
    {
        // تحقق إذا كان العدد زوجي أو فردي
        // Check if number is even or odd
        if (num % 2 == 0)
        {
            evenCount++; // لـ couting الأرقام الزوجية
        }
        else
        {
            oddCount++; // لـ couting الأرقام الفردية
        }
    }

    // عرض النتائج
    // Display results
    Console.WriteLine("\n==== / النتائج / Results ====");
    Console.WriteLine($" عدد الأرقام الزوجية (Even numbers): {evenCount};");
    Console.WriteLine($" عدد الأرقام الفردية (Odd numbers): {oddCount};");

    // انتهي بـ ReadKey()
    // End with ReadKey()
    Console.WriteLine("...اضغط أي مفتاح للخروج\n");
    Console.ReadKey();
}
}

```

شرح الكود خطوة بخطوة / Step-by-Step Explanation:

- إنشاء المصفوفة `numbers = new int[10];`
أرقام زوجية
- متغيرات العد `evenCount` و `oddCount`
لحساب عدد الأرقام الزوجية والفردية
- للإدخال حلقة `for`:
تطلب من المستخدم إدخال رقم
للـ `for`:
- للـ `foreach`:
تمر على كل رقم في المصفوفة
 - (باقي القسمة تتحقق من الزوجي/فرد)
 - فالرقم زوجي 0 إذا كان
 - وإلا فهو فردي
- طبع النتائج الزوجية والفردية

مثال على التشغيل / Example Run:

```

أدخل الرقم 1 : 5
أدخل الرقم 2 : 8
أدخل الرقم 3 : 3
...
عدد الأرقام الزوجية : 4

```

Q3: Answer only four from the following (أجب على أربعة فقط من التالي)

1. The correct definition of array is: التعريف الصحيح (للمصفوفة)

Correct Answer: d. float [] x = new float [3];

Explanation: - هو اسم المصفوفة x - float يعني مصفوفة من نوع [] / الشرح
float بـ 3 عناصر - هذا هو الشكل الصحيح: نوع_البيانات[] اسم_المصفوفة [3] ; [نوع_البيانات][الحجم new

b. خطأ في موضع الأقواس - a. float [3] x; - خطأ في الصيغة - c. float [3] x = new float [3];
float [3] x = new float [] ; خطأ في موضع الأقواس -

2. The correct definition of variables in a method: (التعريف الصحيح للمتغيرات في الدالة)

Correct Answer: c. public static void Sum(params int[] k1, params int[] k2) { }

تسمح بإرسال عدد متغير من المعاملات - يمكن استخدام params نوع [] اسم_المعامل مع مصفوفات - الصيغة الصحيحة params

واحد params لا يسمح باستخدام أكثر من C#، في الحقيقة: Important Note: / ملاحظة مهمة
بشكل عام، فالخيارات params في نفس الدالة. الإجابة الأدق هي: - إذا كان السؤال عن صيغة الأخير في قائمة المعاملات params يوضح الصيغة الصحيحة - لكن عملياً، يجب أن يكون

3. The output on screen of the following codes: ناتج (الكود التالي)

```
for (int i = 0; i < 10; i++)
{
    if (i == 2 || i == 6) { continue; }
    if (i == 4 && i == 8) { break; }
    Console.WriteLine(i);
}
```

Correct Answer: c. 01345789

Step-by-Step Explanation:

i	الشرط الثاني الشرط الأول	Action	الناتج
	First Condition	Second Condition	Output
0	لا (i ≠ 2 و i ≠ 6)	4 == 0 (كاذب)	طباعة 0
1	لا	لا	طباعة 1
2	نعم (i == 2)	-	continue (تخطي)
3	لا	لا	طباعة 3
4		4 == 4 (لا) 8 == 4 (خطأ)	طباعة 4 (لكن 4 == 4)
5	لا	لا	طباعة 5
6	نعم (i == 6)	-	continue (تخطي)
7	لا	لا	طباعة 7
8	لا	4 == 8 (لا (كاذب))	طباعة 8
9	لا	لا	طباعة 9

- ينطوي باقي الكود وبذل جهد للنكرار التالي: **Key Points**: - continue: - break: فلما يتحقق الباقي (break) لهذا (i) مستحيل (continue) لهذا (i) == 4 كلاما يجب أن يكون صحيح (break) لهذا (i) == 4 يعني فلما يتحقق الباقي (break) لهذا (i) مستحيل (continue) لهذا (i) == 4 يحدث أبداً

الناتج النهائي: 01345789

4. The correct definition of construct method for class named A: (A) التعرف الصحيح لدالة النساء (لكلas)

الإجابة الصحيحة / Correct Answer: a. A() { }

الشرح / Explanation:

لها قواعد خاصة وهي **Constructor** (المبني)، **public** (العام) أو **private** (الخاص). يمكن أن تكون الكلمات المفتاحية **public** أو **private** عادة تكون **public** لكن **private**.

```
public class A
{
    // تابع ایجاد کننده کلاس
    public A()
    {
        Console.WriteLine("تم إنشاء كائن من كلاس A");
    }
}
```

Wrong Options: - b. a - {} () دالة البناء لا يكون لها نوع إرجاع - d. void A {} (الكلاس بالضبط البناء لا يكون لها نوع إرجاع

5. The modifier that makes code accessible within the same class, or in a class that is inherited: المعدل الذي يجعل الكود قابل للوصول داخل نفس الكلاس أو في كلاس وارث

الإجابة الصحيحة / Correct Answer: c. protected

الشرح / Explanation:

مستويات وصول رئيسية

المعدل	ن خارج الكلاس و الوصول من نفس الكلاس		
public	نعم ✓	نعم ✓	نعم ✓
private	نعم ✓	لا ✗	لا ✗
protected	نعم ✓	نعم ✓	لا ✗
internal	نعم ✓	نعم (في نفس المشروع) ✓	نعم (في نفس المشروع) ✓

مثال عملي / Practical Example:

```
public class Parent
{
    public int publicVar = 1;      // ✓ ००० ०० ०००००० ०००
    private int privateVar = 2;    // ✗ ००० ०० Parent
    protected int protectedVar = 3; // ✗ ००० Parent ०००००००० ००००००००००
    internal int internalVar = 4; // ०००००० ०० ०००

}

public class Child : Parent
{
    public void TestAccess()
    {
        Console.WriteLine(publicVar);    // ✓ ०००
        // Console.WriteLine(privateVar); // ✗ ०००००० ००० ०० !०००
        Console.WriteLine(protectedVar); // ✓ ००० ००० Child ०००
        Console.WriteLine(internalVar); // ✓ ००००००
    }
}
```

”هليساً لا يطلب الصيحة؟“ نفسك كل الكلاسوا أوي protected لماذا
فقط داخل نفس الكلاس - تسمح بالوصول في كلا الحالتين
- protected (تسمح للجميع (أكثر من المطلوب) (فينطلعكم مكافحة

ملخص الإجابات / Summary of Answers

السؤال	الإجابة
Q1	OOP is a programming paradigm based on objects (explained above)
Q2	Complete C# program with array and foreach loop (code above)
Q3.1	d. float [] x = new float [3];
Q3.2	c. public static void Sum(params int[] k1, params int[] k2)
Q3.3	c. 01345789
Q3.4	a. A() { }
Q3.5	c. protected

نصائح للمبتدئين / Tips for Beginners

- اكتب الكود بنفسك و بلا تكلي بالكودة:
 - Practice coding: Write code yourself, don't just read
- استخدم بيئة تطوير ممتازة للمبتدئين: Visual Studio
 - Use Visual Studio: Excellent IDE for beginners
- غير القمقوش والاطلاق:
 - Try examples: Change values and see results
- تساقر أولاً على أنتم لهم الأخطاء:
 - Read error messages: They help you understand problems
- المتغيرات، الحلقات، الشروط، ثم انتقل بالتأم وأربعين الباينات:
 - Start with basics: Variables, loops, conditions, then move to advanced topics

Prepared by: GitHub Copilot / التاريخ Date: December 6, 2025

Good luck with your studies! / حظاً موفقاً في دراستك