Odpovědi pište na zvláštní odpovědní list s vaším jménem a fotografií. Pokud budete odevzdávat více než jeden list s řešením, tak se na 2. a další listy nezapomeňte podepsat. Do zápatí všech listů vždy napište i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

Společná část pro otázky označené A

Předpokládejte následující program:

```
I1 i1 = new C1();
Console.WriteLine(i1.f());

A1 a1 = (A1) i1;
Console.WriteLine(a1.f());

interface I1 {
  int f();
}

class A1 {
  public int f() { return 1; }
}

class B1 : A1, I1 {
  public int f(int x) => x * 2;
}

class C1 : B1 {
  public new int f() { return 3; }
}
```

Otázka č. 1 (A)

Co uvedený program vypíše na 1. řádku výstupu? Detailně vysvětlete proč – jako součást vysvětlení nakreslete interfacovou tabulku metod pro interface `I1` u tříd `A1`, `B1`, `C1`.

Otázka č. 2 (A)

Co uvedený program vypíše na 2. řádku výstupu? Detailně vysvětlete proč. Vysvětlete, kdo a jak (C# překladač, nebo JIT, nebo strojový kód za runtimu) danou variantu zvolí.

Společná část pro otázky označené B

Předpokládejte následující program:

```
S3? s = new S3 { X = 10 };

if (s is not null) {
  s.Value.QuadrupleX();
  Console.WriteLine(s.Value.X);
}

struct S3 {
  public int X;

  public void QuadrupleX() {
    X *= 4;
  }
}
```

Otázka č. 3 (B)

Co nejpřesněji odhadněte, jaká bude velikost samotné proměnné `s` v bytech. Svůj odhad detailně vysvětlete.

Otázka č. 4 (B)

Co uvedený program vypíše? Detailně vysvětlete proč. Závisí výsledek na tom, zda `Nullable<T>.Value` je vlastnost nebo field?

Otázka č. 5

Co níže uvedený program vypíše? Detailně vysvětlete proč.

```
var day = DayOfWeek.Sunday;
day++;
Console.WriteLine(day);

enum DayOfWeek {
  Monday, Tuesday, Wednesday, Thursday,
  Friday, Saturday, Sunday
};
```

Společná část pro otázky označené C

Předpokládejte následující program:

```
class A6 {
  public int X { get; }

  public A6() {
    X = f();
  }

  public virtual int f() => 1;
}

class B6 : A6 {
  private int _y = 2;

  public override int f() => _y + 10;
}

class C6 : B6 {
  public new virtual int f() => 3;
}

class D6 : C6 {
  public sealed override int f() => 4;
}

class Prg6 {
  public static void Main() {
    var d = new D6();
    Console.WriteLine(m1(d));
    Console.WriteLine(d.X);
  }

  public static int m1(A6 a)
    => m2((C6) a);

  public static int m2(C6 c)
    => c.f();
}
```

Otázka č. 6 (C)

Co uvedený program vypíše na 1. řádku výstupu? Detailně vysvětlete proč – jako součást vysvětlení nakreslete VMT tříd `A6`, `B6`, `C6`, `D6`.

Otázka č. 7 (C)

Co uvedený program vypíše na 2. řádku výstupu? Detailně vysvětlete proč.

Otázka č. 8

Co níže uvedený program vypíše? Detailně vysvětlete proč. Vysvětlete, jaký je rozdíl mezi metodami `m1` a `m2`.

```
class Prg8 {
  public static void Main() {
    var items = new List<int>();

    m1(items);
    Console.WriteLine(items.Count);

    m2(ref items);
    Console.WriteLine(items.Count);
  }

  static void m1(List<int> ints) {
    ints.Add(6);
  }

  static void m2(ref List<int> ints) {
    ints.Add(7);
  }
}
```

Otázka č. 9

Níže najdete dokumentaci metody `Sin` standardní .NET třídy `Math`. Předpokládejte, že metodu `Math.Sin` standardní knihovny implementujeme a v rámci implementace k ní chceme napsat sadu unit testů – **s využitím testovacího frameworku** MSTest, xUnit nebo nUnit napište sadu vhodných a relevantních unit testů testujících alespoň 3 rozdílné aspekty chování metody `Math.Sin`. Soustřeďte se na přehlednost testů a jejich dobré pojmenování.

Math.Sin(Double) Method

Definition

Namespace: System
Assembly: System.Runtime.dll
Returns the sine of the specified angle.

```
public static double Sin (double a);
```

Parameters

a Double
An angle, measured in radians.

Returns

Double

The sine of a. If a is equal to NaN, NegativeInfinity, or PositiveInfinity, this method returns NaN.

Otázka č. 10

Předpokládejte níže uvedenou třídu `WebServer` pro zpracování požadavků na webovém serveru specifickém pro dodavatele náhradních dílů pro důlní stroje. Očekáváme velký počet požadavků na server, proto počítáme s paralelním zpracování požadavků ve více vláknech. Vysvětlete, proč uvedený objektový návrh není vhodný, a upravte ho na vhodnější (úpravou stávajícího kódu, přidáním nových typů, atd.). Vysvětlete, v čem je váš objektový návrh lepší.

```
class WebServer {
  public static bool DebugMode;

  public static string HandleRequest(string path) {
    var html = new StringBuilder();
    var tokens = path.Split('/');
    switch (tokens[0]) {  // Handle all pages
      case "invoice":
        switch (tokens[1]) {
          // Handle all invoice subpages
          case "list":
            // Generate invoice list HTML → html
            if (DebugMode) {
              // Append list specific logs → html
            }
            break;
          case "approval":
            // Gen. invoice approval HTML → html
            if (DebugMode) {
              // Append approval spec. logs → html
            }
            break;
          default:
            int id = int.Parse(tokens[1]);
            // Generate invoice "id" HTML → html
            if (DebugMode) {
              // Append invoice "id" spec. logs
            }
            break;
        }
        break;
      // TODO: Implement further pages and subpages
    }
    return html.ToString();
  }
}
```