コース名：	VB プログラミング基礎

問題 1

正解：2　　［配点：4 点］

解説

VB や C# などの.NET Framework 対応言語の特徴の一つとして、イベントドリブンが挙げられます。.NET Framework とは、さまざまな.NET アプリケーションを開発、実行するためのマルチ言語環境で、開発マシンと実行マシンの両方にインストールする必要があります。VB や C#といったプログラミング言語の種類に依存せずにアプリケーションの開発、実行ができるため、言語ごとの個別のクラスライブラリを用意する必要はありません。

「参考ページ」 1 章 3, 4, 5, 8 ページ

問題 2

正解：1　　［配点：4 点］

解説

CLR（Common Language Runtime：共通言語ランタイム）は、.NET Framework 上でアプリケーションやコンポーネントを動作させるための実行エンジンであり、中間言語（MSIL）を実行するための環境です。名前空間は、.NET Framework クラスライブラリを機能別に分類したものです。クラスライブラリは、.NET Framework の CLR 環境上で提供されるプログラムの部品群です。また、Windows フォームを使用して Windows アプリケーションを作成できます。

「参考ページ」 1 章 7, 8, 9 ページ

問題 3

正解：2　　［配点：4 点］

解説

VB のソースコードは、VB コンパイラにより MSIL（Microsoft Intermediate Language）に変換されます。コンパイルされたファイルは、実行ファイル（exe や dll）として生成されます。その後、実行時に JIT コンパイラにより、実行プラットフォームに適したネイティブコードに変換されます。

「参考ページ」 1 章 10 ページ

問題 4

正解：1　　［配点：4 点］

解説

変数は、定義と同時に初期値を設定できます。初期値とデータ型は省略できます。また、1 つのステートメントで同じデータ型の変数をカンマ区切りで複数定義できます。変数名は大文字小文字を区別しないため、Name と name は同じ変数です。
「参考ページ」 2 章 27 ページ

問題 5

正解：3　　［配点：4 点］

解説

数値型の値を String 型に変換するには、ToString メソッドを使用します。変換する書式は、「数値型データ.ToString()」 です。本問では、数値型データは number なので message = number.ToString() が正しい記述となります。
「参考ページ」2 章　34 ページ

問題 6

正解：4　　［配点：4 点］

解説

配列は同じデータ型の要素を複数扱えます。配列の各要素を操作するには、Index 番号を使用します。配列の Index 番号は 0 から始まります。配列の定義と同時に、要素に初期値を設定できます。また、Length プロパティを使用すると、配列の要素数を取得できます。
「参考ページ」 2 章 36, 37 ページ

問題 7

正解：2　　［配点：4 点］

解説

動的配列は、受け取る値の数に応じて自由に要素数を変更できる配列です。そのため、通常の配列と違い、Index 番号や初期値を指定しなくても動的配列を定義できます。定義時に指定したデータ型の値であれば、Add メソッドや RemoveAt メソッドを使用して、要素数をいくつでも追加・削除できます。なお、左辺の As List(Of データ型) は省略できます。
「参考ページ」 2 章　39, 40, 41, 42 ページ

問題 8

正解：2　　［配点：4 点］

解説

Do While…Loop は前判定の繰り返しを行う制御構文です。今回は、変数 number が 10 より小さいという条件が成り立つ間、処理を繰り返します。処理では、number と total の値を加算した結果を変数 total に代入した後、number に 2 を加算します。したがって、number は 2、4、6、8 と増加し、実行結果は「20」と表示されます。

「参考ページ」 2 章 48 ページ

問題 9

正解：4　　［配点：4 点］

解説

配列などのコレクションの各要素に同様の処理を繰り返すときには、In キーワードの後に繰り返しの対象となるコレクションを指定します。コレクションとは、関連するデータ型をグループ化したものです。配列もコレクションの一種です。In の前には、反復変数を定義します。反復変数は事前に定義することもできます。コレクションの各要素を 1 つずつ反復変数で受け取り、繰り返し処理を実行します。

「参考ページ」 2 章　51 ページ

問題 10

正解：2　　［配点：4 点］

解説

オブジェクト指向プログラミングでは、現実世界の「もの」に着目してプログラムを記述します。また、オブジェクトの持つデータをフィールド、処理をメソッドと呼びます。オブジェクト指向プログラミングでは、ソフトウェアの拡張性や再利用性を改善し、ソフトウェアの保守を容易かつ低コスト化することを目標にしています。なお、オブジェクト指向プログラミングでは、オブジェクトのひな形をクラスとしてプログラムを記述します。

「参考ページ」 3 章 61, 62 ページ

問題 11

正解：3　　［配点：4 点］

解説

カプセル化は、オブジェクトのフィールドにアクセスする方法をメソッド経由に制限し、データを保護することで、データの安全性を高めます。継承は、オブジェクトを再利用する方法の一つです。既存のクラス定義を引き継ぐため、新しいクラスには差分のメンバーのみを追加します。継承のもとになるクラスを基本クラス、継承して作られたクラスを派生クラスと呼びます。ポリモーフィズムは、異なるオブジェクトに対して同じ名前のメソッドやプロパティを使用してアクセスできる仕組みのことです。
「参考ページ」 3 章　65, 66, 67, 68 ページ

問題 12

正解：4　　［配点：4 点］

解説

クラス内に定義するメンバーのうち、外部に公開するものには Public、外部に公開せずクラス内でのみ使用するには Private のアクセス修飾子を指定します。Friend のアクセス修飾子は、定義したクラスと同一アセンブリ内からのみ使用できます。Protected のアクセス修飾子は定義したクラスの内部、および派生クラスの内部からのみ使用できます。
「参考ページ」 4 章 74 ページ

問題 13

正解：3　　［配点：4 点］

解説

ReadOnly 修飾子を付けると、Get アクセサー（Get～End Get）のみを作成し、WriteOnly 修飾子を付けると、Set アクセサー（Set～End Set）のみを作成します。また、 Property キーワードに続けて指定した名前がプロパティ名として使用されるため、Get アクセサー、Set アクセサーでそれぞれプロパティ名をつけることはできません。
「参考ページ」 4 章 76, 78 ページ

問題 14

正解：4　　［配点：4 点］

解説

クラスからインスタンスを生成する場合、オブジェクト変数の定義と New 演算子によるインスタンスの生成が必要です。今回は、Item クラスにデフォルトコンストラクターではなく String 型の引数を 1 つ受け取るコンストラクターのみが定義されているため、任意の文字列（"apple"）を引数としてインスタンスを生成します。
「参考ページ」 4 章 80, 94 ページ

問題 15

正解：2　　［配点：4 点］

解説

オーバーロードとは、同じ名前のメソッドを複数定義できる機能です。オーバーロードできる条件は、定義するメソッドの引数の数や型、並び順が異なることです。引数の名前や戻り値の型だけが異なる場合はオーバーロードできません。
「参考ページ」 4 章　93 ページ

問題 16

正解：1　　［配点：4 点］

解説

Function メソッドは、戻り値を呼び出し元のメソッドに返します。戻り値を返すには、Return キーワードのあとに戻り値を指定します。
「参考ページ」 4 章 87 ページ

問題 17

正解：3　　［配点：4 点］

解説

コンストラクターは、インスタンスが生成される際に自動的に呼び出される特殊なメソッドです。フィールドの初期化などの役割を持ち、オーバーロードできます。クラス内にコンストラクターを 1 つも定義しなかった場合は、引数のないデフォルトコンストラクターがコンパイラによって作成されます。
「参考ページ」 4 章 94 ページ

問題 18

正解：2　　［配点：4 点］

解説

派生クラスを定義するには、派生クラス側で Inherits キーワードを使用します。派生クラスには基本クラスのメンバーがそのまま引き継がれるので、基本クラスのメンバーを再利用できます。派生クラスには追加で必要なフィールドやメソッドのみを定義します。派生クラスから基本クラスのメンバーを呼び出すには、MyBase キーワードを使用します。
「参考ページ」 5 章 111, 113 ページ

問題 19

正解：2　　［配点：4 点］

解説

基本クラスのメソッドを派生クラスで再定義することをオーバーライドといいます。オーバーライドを実装するには、基本クラスのメソッドに Overridable 修飾子、派生クラスのメソッドに Overrides 修飾子を使用します。MyBase キーワードは、オーバーライドされた基本クラスのメソッドを呼び出すために使用します。

「参考ページ」5 章　113 ページ

問題 20

正解：3　　［配点：4 点］

解説

抽象クラスの定義には、MustInherit キーワードを使用します。抽象クラスは具体的な処理を持たない抽象メソッドだけでなく、一般的なメソッドを定義できます。ただし、抽象クラスからはインスタンスを生成できません。インスタンスを生成するために、必ず派生クラスを定義します。派生クラスでは抽象メソッドをすべてオーバーライドして再定義します。

「参考ページ」6 章　123, 124 ページ

問題 21

正解：1　　［配点：4 点］

解説

インターフェイスには空のメソッドのみ定義できます。Function メソッドだけでなく、Sub メソッドも定義できます。インターフェイスは継承と異なり、1 つのクラスに複数実装できます。定義したインターフェイスを実装する場合は、Implements キーワードを使用します。

「参考ページ」6 章　126, 127, 128 ページ

問題 22

正解：3　　［配点：4 点］

解説

例外が発生すると、例外と一致する型を定義している Catch ブロックが例外を受け取ります。本問では、String 型から Integer 型への変換時の例外に対処するため、ランタイム例外の FormatException を Catch ブロックに記述しておきます。TextBox1 に入力された任意の文字列（aaa）を Integer 型へ変換する処理を実行すると、実行時エラーが発生します。すると、FormatException 例外クラスから例外オブジェクトが生成され返されます。

「参考ページ」　7 章　146, 147 ページ

問題 23

正解：4　　［配点：4 点］

解説

Catch ブロックに Throw キーワードを記述すると呼び出し元に例外を伝播できます。また、キャッチした例外の再スローも可能です。ただし、数値や文字列などを直接スローすることはできません。Try ブロックでは Throw New 例外名([引数リスト])と記述するとユーザー定義例外を発生させることができるため、ユーザーが定義したオリジナルの例外にも対応できます。
「参考ページ」 7 章 149, 150 ページ

問題 24

正解：3　　［配点：4 点］

解説

ブレークポイントは、特定の位置でプログラムの実行を一時停止し、中断モードに切り替えるための設定です。プログラムを実行していない状態でも、ブレークポイントを設定、解除できます。ブレークポイントは複数行に設定できます。
「参考ページ」 8 章 158 ページ

問題 25

正解：2　　［配点：4 点］

解説

ブレークポイントでプログラムの実行を中断し、1 行ずつコードを実行するには、ステップ実行をします。ステップインは呼び出し先のメソッドの処理を 1 行ずつ実行します。ステップアウトは現在の黄色いハイライト行が存在するメソッドの残りの行を一括実行します。ステップオーバーは呼び出し先のメソッドを一括実行します。 ウォッチウィンドウは変数と式を評価するために使用します。
「参考ページ」 8 章 159, 160 ページ