SOLUTIONS

This document contains the solutions to selected Try It Out exercises and the Programming Challenges at the end of each chapter.

Chapter 1

Programming Challenge 1-1:

- Object: `TextWindow`
- Method: `WriteLine()`
- Arguments: `"Today is Friday."` and `"I lost track of what day it is."`
- Keywords: `If`, `Then`, `Else`, and `EndIf`

Programming Challenge 1-2: Here is a sample solution. Use your own name.

```
' Prob_1_2.sb
TextWindow.WriteLine("Hello Majed!")
```

Programming Challenge 1-3: The `ShowMessage()` method takes two arguments. The first argument is the text to be displayed in the message box, and the second argument is the title for the message box.

```
' Prob_1_3.sb
GraphicsWindow.ShowMessage("Hello Majed!", "Greetings")
```

Chapter 2

Try It Out 2-1:

Problem	Object	Method	Arguments
1	Shapes	AddRectangle()	100, 50
2	Math	Max()	5, 10
3	Sound	PlayBellRing()	

Try It Out 2-2: See the file *TryItOut_2_2.sb*.

Try It Out 2-3: See the file *TryItOut_2_3.sb*.

Try It Out 2-4: See the file *TryItOut_2_4.sb*.

Programming Challenge 2-1: Answers will vary. See the file *Prob_2_1.sb* for a sample solution.

Programming Challenge 2-2: Answers will vary. See the file *Prob_2_2.sb* for a sample solution.

Programming Challenge 2-3: Cathy forgot the plus (+) sign (the concatenation operator). See the file *Prob_2_3.sb* for the correct program.

Programming Challenge 2-4: Answers will vary. See the file *Prob_2_4.sb* for a sample solution.

Chapter 3

Try It Out 3-1: See the file *TryItOut_3_1.sb*.

Try It Out 3-2: See the file *TryItOut_3_2.sb*.

Try It Out 3-3: See the file *TryItOut_3_3.sb*.

Programming Challenge 3-1: You'll get a star shape. See the file *Prob_3_1.sb*.

Programming Challenge 3-2: The code draws a star. See the file *Prob_3_2.sb*.

Programming Challenge 3-3: The code draws a computer and monitor. See the file *Prob_3_3.sb*.

Programming Challenge 3-4: Answers will vary.

Programming Challenge 3-5: See the file *Prob_3_5.sb*.

Programming Challenge 3-6: See the file *Prob_3_6.sb*.

Programming Challenge 3-7: See the file *Prob_3_7.sb*.

Programming Challenge 3-8: See the file *Prob_3_8.sb*.

Programming Challenge 3-9: See the file *Prob_3_9.sb*.

Programming Challenge 3-10: See the file *Prob_3_10.sb*.

Programming Challenge 3-11: See the file *Prob_3_11.sb*.

Programming Challenge 3-12: See the file *Prob_3_12.sb*.

Chapter 4

Try It Out 4-1: The program has three variables: `mathHours`, `scienceHours`, and `avgHours`. The program's output is shown below:

```
I spend 8 hours a week on math homework and 6 hours a week on science
homework.

The average of 8 and 6 is 7.
```

Try It Out 4-2: The variable `1MoreRound` is invalid because it starts with a number. The variable `$FinalScore` is also invalid because its starts with a $. The other two variables (`_myBooK` and `Level2`) are valid.

For the second part, answers will vary. Here are some suggestions:

- The score of a player in a game: `score` or `playerScore`
- The hypotenuse of a right triangle: `hyp` or `hypotenuse`
- The number of floors in a building: `numFloors` or `floorCount`
- The number of miles a car can drive per gallon of fuel: `milesPerGal` or `mpg`
- The number of licks it takes to get to the center of a Tootsie Pop: `numLicks` or `licksToCenter`

Try It Out 4-3: See the file *TryItOut_4_3.sb*.

Try It Out 4-4: See the file *TryItOut_4_4.sb*.

Try It Out 4-5: This is what you get when you run this program:

```
Before: x =  and y =
After: x = 10 and y = 10
```

When the first statement in the program is run, the variables `x` and `y` are empty (they haven't been initialized yet). This is why the first `WriteLine()` call displays nothing (an empty string) for these variables.

Programming Challenge 4-1: See the file *Problem_4_1.sb*. To tell a different joke, you just need to assign different strings to the `name` and `reply` variables.

Programming Challenge 4-2: See the file *Problem_4_2.sb*.

Chapter 5

Try It Out 5-1: See the file *TryItOut_5_1.sb*.

Try It Out 5-2: See the file *TryItOut_5_2.sb*.

Try It Out 5-3: See the file *TryItOut_5_3.sb*.

Try It Out 5-4: See the file *TryItOut_5_4.sb*.

Try It Out 5-5: See the file *TryItOut_5_5.sb*.

Programming Challenge 5-1: See the file *Prob_5_1.sb*.

Programming Challenge 5-2: See the file *Prob_5_2.sb*.

Programming Challenge 5-3: See the file *Prob_5_3.sb*.

Chapter 6

Try It Out 6-1: See the file *TryItOut_6_1.sb*.

Try It Out 6-2: See the file *TryItOut_6_2.sb*.

Programming Challenge 6-1: The answer is always 8 regardless of the number you pick. Let's assume that your number is x and use a little algebra to follow the magician's instructions:

a. Your secret number: x
b. Subtract 1 from your number: $x - 1$
c. Multiply the result by 3: $3\ (x - 1)$
d. Add 12 to the result: $3(x - 1) + 12 = 3x - 3 + 12 = 3x + 9$
e. Divide the answer by 3: $(3x + 9) \div 3 = x + 3$
f. Add 5 to the answer: $x + 8$
g. Subtract your secret number: $(x + 8) - x = 8$

Programming Challenge 6-2: See the file *Prob_6_2.sb*.

Programming Challenge 6-3: See the file *Prob_6_3.sb*.

Chapter 7

Try It Out 7-1: See the file *TryItOut_7_1.sb*.

Try It Out 7-2: See the file *TryItOut_7_2.sb*.

Try It Out 7-3: See the file *TryItOut_7_3.sb*.

Try It Out 7-4: See the file *TryItOut_7_4.sb*.

Try It Out 7-5: See the file *TryItOut_7_5.sb*.

Try It Out 7-6: See the file *TryItOut_7_6.sb*.

Programming Challenge 7-1:

a.
```
a = Math.Pi * r * Math.SquareRoot(r * r + h * h)
```

b.
```
a = Math.Power(x, Math.Power(y, z))
```

c.
```
a = Math.SquareRoot((x + y) / z)
```

Programming Challenge 7-2: See the file *Prob_7_2.sb*.

Programming Challenge 7-3: See the file *Prob_7_3.sb*.

Chapter 8

Try It Out 8-1: See the file *TryItOut_8_1.sb*.

Try It Out 8-2: See the file *TryItOut_8_2.sb*.

Try It Out 8-3: See the file *TryItOut_8_3.sb*.

Try It Out 8-4: See the file *TryItOut_8_4.sb*.

Try It Out 8-5: See the file *TryItOut_8_5.sb*.

Programming Challenge 8-1: See the file *Prob_8_1.sb*.

Programming Challenge 8-2: See the file *Prob_8_2.sb*.

Programming Challenge 8-3: See the file *Prob_8_3.sb*.

Programming Challenge 8-4: See the file *Prob_8_4.sb*.

Chapter 9

Try It Out 9-1: See the file *TryItOut_9_1.sb*.

Try It Out 9-2: See the file *DiceGame.sb* in the folder *TryItOut_9_2*.

Try It Out 9-3: Answers will vary.

Try It Out 9-4: See the file *TryItOut_9_4.sb*.

Programming Challenge 9-1: See the file *Prob_9_1.sb*.

Programming Challenge 9-2: See the file *HungryMouse.sb* in the folder *Prob_9_2*.

Programming Challenge 9-3: See the file *Prob_9_3.sb*.

Chapter 10

Try It Out 10-1: See the file *TryItOut_10_1.sb*.

Try It Out 10-2: See the file *TryItOut_10_2.sb*.

Try It Out 10-3: Answers will vary.

Try It Out 10-4: Answers will vary.

Programming Challenge 10-1: See the file *Monster.sb* in the *Prob_10_1* folder.

Programming Challenge 10-2: See the file *GhostHunt.sb* in the *Prob_10_2* folder.

Chapter 11

Try It Out 11-1: See the file *TryItOut_11_1.sb*.

Try It Out 11-2: See the file *TryItOut_11_2.sb*. You should see that the `KeyDown` event is raised continuously as long as a key is pressed.

Try It Out 11-3: See the file *TryItOut_11_3.sb*.

Try It Out 11-4: See the file *TryItOut_11_4.sb*.

Try It Out 11-5: Answers will vary.

Programming Challenge 11-1: See the file *Prob_11_1.sb*.

Programming Challenge 11-2: See the file *Prob_11_2.sb* in the folder *Prob_11_2*.

Programming Challenge 11-3: Answers will vary. See the file *Maze.sb* in the folder *Prob_11_3* for a sample solution.

Chapter 12

Try It Out 12-1: Answers will vary.

Try It Out 12-2: See the file *TryItOut_12_2.sb*.

Try It Out 12-3: Answers will vary.

Try It Out 12-4: See the file *TryItOut_12_4.sb*.

Programming Challenge 12-1: See the file *HiddenTreasure.sb* in the folder *Prob_12_1*.

Programming Challenge 12-2: See the file *SeaWorld.sb* in the folder *Prob_12_2*.

Chapter 13

Try It Out 13-1: Answers will vary.

Try It Out 13-2: See the file *TryItOut_13_2.sb*.

Try It Out 13-3: See the file *TryItOut_13_3.sb*. Note that 1 + 100 = 101, 2 + 99 = 101, 3 + 98 = 101, . . . 50 + 51 = 101. So the answer is simple: 101 × 50 = 5050.

Try It Out 13-4: See the file *TryItOut_13_4.sb*.

Try It Out 13-5: See the file *TryItOut_13_5.sb*. The program draws 200 lines from the upper-left corner to random points in the graphics window.

Try It Out 13-6: See the file *TryItOut_13_6.sb*.

Try It Out 13-7: Answers will vary.

Try It Out 13-8: See the file *TryItOut_13_8.sb*.

Programming Challenge 13-1: See the file *Prob_13_1.sb*.

Programming Challenge 13-2: See the file *Prob_13_2.sb*.

Programming Challenge 13-3: See the file *Prob_13_3.sb*.

Programming Challenge 13-4: See the file *Prob_13_4.sb*.

Chapter 14

Try It Out 14-1: See the file *Woodchuck.sb*. Answers will vary for the improvements.

Try It Out 14-2: See the file *TryItOut_14_2.sb*.

Try It Out 14-3: See the file *TryItOut_14_3.sb*.

Try It Out 14-4: Answers will vary.

Programming Challenge 14-1: See the file *Race.sb* in the folder *Prob_14_1*.

Programming Challenge 14-2: See the file *SimpleSlot.sb* in the folder *Prob_14_2*.

Programming Challenge 14-3: See the file *Space.sb* in the folder *Prob_14_3*. Answers will vary for the improvements.

Chapter 15

Try It Out 15-1: See the file *TryItOut_15_1.sb*.

a. `S[A] = 3.5`

b. `S[B] = 2`

c. `S[A*B-2] = -1`

d. `S[A+B] = 6`

e. `S[A]-2*S[B] = -0.5`

Try It Out 15-2: See the file *TryItOut_15_2.sb*. The file shows three ways to solve the problem.

Try It Out 15-3: See the file *TryItOut_15_3.sb*.

Try It Out 15-4: See the file *TryItOut_15_4.sb*.

Try It Out 15-5: See the file *TryItOut_15_5.sb*.

Try It Out 15-6: Answers will vary.

Programming Challenge 15-1: See the file *Dice.sb* in the folder *Prob_15_1*.

Programming Challenge 15-2: See the file *PinBall.sb* in the folder *Prob_15_2*.

Programming Challenge 15-3: See the file *FlowerAnatomy.sb* in the folder *Prob_15_3*.

Programming Challenge 15-4: See the file *USMapQuiz.sb* in the folder *Prob_15_4*.

Chapter 16

Try It Out 16-1: If the user enters an invalid name, then `day[name]` is empty and the program won't display anything after the word "is". See the file *TryItOut_16_1.sb*.

Try It Out 16-2: Answers will vary.

Try It Out 16-3: Check the file *AnimalSpeed.sb* in the folder *TryItOut_16_3*.

Try It Out 16-4: Answers will vary.

Programming Challenge 16-1: See the file *Prob_16_1.sb*.

Programming Challenge 16-2: See the file *Prob_16_2.sb*.

Programming Challenge 16-2: See the file *VirtualPiano.sb* in the folder *Prob_16_3*.

Chapter 17

Try It Out 17-1: See the file *TryItOut_17_1.sb*.

Try It Out 17-2: See the file *TryItOut_17_2.sb*.

Try It Out 17-3: See the file *TryItOut_17_3.sb*.

Try It Out 17-4: See the file *TryItOut_17_4.sb*.

Try It Out 17-5: See the file *TryItOut_17_5.sb*.

Try It Out 17-6: See the file *TryItOut_17_6.sb*.

Try It Out 17-7: Answers will vary.

Programming Challenge 17-1: See the file *Okla.sb* in the folder *Prob_17_1*.

Programming Challenge 17-2: See the file *TicTacToe.sb* in the folder *Prob_17_2*.

Chapter 18

Try It Out 18-1: See the file *TryItOut_18_1.sb*.

Try It Out 18-2: See the file *TryItOut_18_2.sb*.

Try It Out 18-3: See the file *TryItOut_18_3.sb*.

Try It Out 18-4: The code point for character *A* is 65. The code runs a loop that changes `code` from 65 to 90. In each iteration, the code uses `Text.GetCharacter(code)` to get the character that corresponds to the current value of `code`, and then displays that character on a new line.

Try It Out 18-5: See the file *TryItOut_18_5.sb*.

Try It Out 18-6: See the file *TryItOut_18_6.sb*.

Try It Out 18-7: See the file *TryItOut_18_7.sb*.

Try It Out 18-8: See the file *TryItOut_18_8.sb*.

Try It Out 18-9: See the file *TryItOut_18_9.sb*.

Try It Out 18-10: Answers will vary.

Programming Challenge 18-1: See the file *Shoot.sb* in the folder *Prob_18_1*.

Programming Challenge 18-2: See the file *BinaryToDecimal.sb* in the folder *Prob_18_2*.

Chapter 19

Try It Out 19-1: See the file *TryItOut_19_1.sb*.

Try It Out 19-2: See the file *TryItOut_19_2.sb*.

Try It Out 19-3: Answers will vary.

Programming Challenge 19-1: See the file *Homonyms.sb* in the folder *Prob_19_1*.

Programming Challenge 19-2: See the file *AnimalKingdom.sb* in the folder *Prob_19_2*.