Chapter Review and Exercises

Key Terms

`break;`
compound conditions
counters
curly brackets ({})
dual-alternative structure
`if...`
`if...else`
`if...else if...`
`Math` object
`Math.floor()` method
`Math.random()` method
multiple-alternative structure
nested selection structure
selection structure
`return;` statement
single-alternative structure
`switch` statement
test condition

Review Exercises

Fill in the Blank

1. A selection structure needs a _____ _____ along with one or more blocks of statements.
2. An `if...else` clause is an example of a _____-alternative structure.
3. Curly brackets are not required in an `if...` clause if the clause contains only _____ _____.
4. A _____ statement can often be used to replace several `if...else if...` structures.
5. A _____ is used to keep track of how many times something has happened in a program.

True or False

6. T/F An `if...else` structure needs only one test condition.
7. T/F A multiple-alternative selection structure needs only one test condition.
8. T/F The only possible values for a test condition are `true` and `false`.
9. T/F Curly brackets must always be used to enclose statements in an `if...` clause even if there is only one statement.
10. T/F The `return;` statement will cause a break in the execution of a clause in a selection structure.
11. T/F An `if...else` structure cannot be nested within another `if...else` structure.
12. T/F The test condition of an `else...` clause must be the same as the test condition of its `if...` clause.
13. T/F All cases in a `switch` statement *must* include a `break;` statement.

14. T/F If the test condition in an `if...` clause is `false`, the statements in that clause will be skipped.

15. T/F A test condition can never contain compound conditions.

Short Answer

16. What is the result of the following statement, given that `x = 4`?

```
if(x == 5)
```

17. What will be displayed after the following code snippet is run, given that `Jody = 18`?

```
if(Jody > 18)
   document.write("You can vote");
if(Jody < 18)
   document.write("You're too young");
document.write("Bye bye");
```

18. What will be displayed after the following code snippet is run, given that `Jody = 18`?

```
if(Jody > 18)
   document.write("You can vote");
else if(Jody < 18)
   document.write("You're too young");
document.write("Bye bye");
```

19. What will be displayed after the following code snippet is run, given that `rain = "yes"`?

```
if(rain == "yes")
   document.write("Bring your umbrella");
else
{
   document.write("No umbrella needed");
   document.write("Bye bye");
}
```

20. What will be displayed after the following code snippet is run, given that `rain = "yes"`?

```
if(rain == "yes")
   document.write("Bring your umbrella");
else
   document.write("No umbrella needed");
   document.write("Bye bye");
```

21. A `switch` statement is often used to replace which of the following?
    a) Multiple `if...` statements
    b) A single `if...else if...` structure
    c) Multiple `if...else if...` statements
    d) Only (a) and (c)
    e) Any of the following: (a), (b), or (c)

22. What value is begin tested in the following code snippet?

```
switch(points)
{
      case 20:
           document.write("twenty");
           break;
      case 20:
           document.write("twenty");
           break;
      default:
           document.write("invalid entry");
}
```

23. Assume you want a user to enter a number between 10 and 20 (inclusive). Write the compound condition that should be entered in the following `if...` clause that will test for a number in this range, using the AND operator:

```
var num = parseInt(prompt("enter a number between 10 and 20"," "));
if(_____________________________)
   document.write("good number");
else
   document.write("invalid number");
```

24. Assume you want a user to enter a number between 10 and 20 (inclusive). Write the compound condition that should be entered in the following `if...` clause that will test for a number in this range, using the OR operator:

```
var num = parseInt(prompt("enter a number between 10 and 20"," "));
if(_____________________________)
   document.write("invalid number");
else
   document.write("good number");
```

25. Assume you want a user to enter a number between 10 and 20 (inclusive). Write the compound condition that should be entered in the following `if...` clause that will test for a number in this range, using the OR and NOT operators:

```
var num = parseInt(prompt("enter a number between 10 and 20"," "));
if(_____________________________)
   document.write("good number");
else
   document.write("invalid number");
```

26. What will be displayed if the following code snippet is run, given that the user enters `apple` at the prompt?

```
var fruit = prompt("What do you want to eat? ", " ");
   switch (fruit)
   {
      case "apple":
        document.write("An apple a day is good for you.<br />");
      case "banana":
        document.write("Bananas are delicious.<br />");
      case "grapes":
        document.write("Who doesn't like grapes!<br />");
      default:
        document.write("You need to eat more fruit.<br />");
   }
```

27. Fix the code in Exercise 26 so that only the response that corresponds with the user's entry displays.

28. Add an `if...` clause to the program snippet you fixed in Exercise 27 so that, if the user enters `apple`, the program will prompt the user to enter a variety of apple, such as Granny Smith or McIntosh, and the output will use that apple variety to say `"A _____ apple is good for you."`

Exercises 29 and 30 refer to the following code:

```
var vacation = prompt("What do you want to do during Spring Break? ↵
                Type S for skiing, F for fishing, H for hiking,&
                J for learning JavaScript.", " ");
if (vacation == 'S')
   document.write("You should go to Aspen.<br />");
if (vacation == 'H')
   document.write("Be sure to buy good hiking boots.<br />");
if (vacation == 'F')
   document.write("Worms make good bait.<br />");
if (vacation == 'J')
   document.write("You're gonna have soooo much fun!<br />");
```

29. Rewrite the code as a series of `if...else if...` structures.

30. Rewrite the code using a `switch` statement.

Programming Challenges

On Your Own

1. Create a web page that acts as a temperature converter. The user should be given the option to enter the temperature in degrees Fahrenheit and the program will convert the temperature to degrees Celsius. Alternatively, the user can enter the temperature in degrees Celsius and the program will convert the temperature to degrees Fahrenheit. Save the page as `temp.html` and be sure to include an appropriate page title. The formulas for the conversions are as follows:

   - `Celsius = 5/9 * (Fahrenheit - 32)`
   - `Fahrenheit = (Celsius * 5/9) + 32`

2. Either create a new web page or add to the page you created in Programming Challenge 1. The page will act as a weather forecaster. The user will enter a temperature in either degrees Celsius or degrees Fahrenheit (your choice) and the program will display one of the following messages, depending on the temperature:

   - Less than 0^0F or -18^0C: `Bundle up! It's really freezing out there!`
   - $0^0 - 32^0$F or $-18^0 - 0^0$C: `Pretty cold with a chance of snow.`
   - $33^0 - 59^0$F or $-17^0 - 15^0$C: `Don't forget your jacket. It's still chilly outside.`
   - $60^0 - 80^0$F or $16^0 - 27^0$C: `Perfect lovely weather...unless it rains.`
   - $81^0 - 95^0$F or $28^0 - 35^0$C: `Nice and warm. Go for a swim.`

- Greater than 95^0F or 35^0C: `It's really hot! Probably best to stay in an air conditioned spot.`

Save your page with the filename `forecast.html` and be sure to include an appropriate page title.

ideoNote
sing the Switch Structure
n_Your_Own_3_Grades

3. Create a web page that converts a student's course average to a letter grade. Save your page as `grades.html` and be sure to include an appropriate page title. The conversions are as follows:
   - Less than 60: `F`
   - 60 – 69.5: `D`
   - 69.6 – 79.5: `C`
   - 79.6 – 89.5: `B`
   - Greater than 89.5: `A`
4. Create a web page that calculates an employee's net pay. The program should prompt for an employee's hourly pay rate, number of hours worked per week, and number of dependents claimed. If the employee works more than 40 hours in a week, overtime is calculated at 1.5 times the regular hourly rate. Taxes are then deducted from the gross pay as follows:
   - No dependents: tax rate is 28%
   - 1 to 3 dependents: tax rate is 25%
   - 4 to 6 dependents: tax rate is 15%
   - More than 6 dependents: tax rate is 10%

   Save your page as `paychecks.html` and be sure to include an appropriate page title.
5. Create a web page like the one created in Programming Challenge 4. However, this program will calculate taxes based on compound conditions as follows:
   - No dependents and gross pay is greater than $1000.00: tax rate is 33%
   - No dependents and gross pay is less than or equal to $1000.00: tax rate is 28%
   - 1 to 3 dependents and gross pay is greater than $1000.00: tax rate is 25%
   - 1 to 3 dependents and gross pay is less than or equal to $1000.00: tax rate is 22%
   - 4 to 6 dependents and gross pay is greater than $1000.00: tax rate is 22%
   - 4 to 6 dependents and gross pay is less than or equal to $1000.00: tax rate is 15%
   - More than 6 dependents and gross pay is greater than $1000.00: tax rate is 15%
   - More than 6 dependents and gross pay is less than or equal to $1000.00: tax rate is 10%

   Save your page as `paychecks2.html` and be sure to include an appropriate page title.
6. Create a web page for a game site that allows the player to "purchase" various items, depending on the points the player amasses. The player should be

prompted to enter the number of points he or she has and then be allowed to select an item to purchase. If the player has enough points to cover the cost, a message will inform the player that the item has been added to his or her inventory. If the player does not have sufficient points to purchase the item, a message will appear saying that no purchase can be made. The items to be purchased should include the following but you can add your own items as well: a sword, a water skin to hold a gallon of water, a charm which allows the user to become invisible for five minutes, and a cell phone.

Save your page as `points.html` and be sure to include an appropriate page title.

7. Create a web page that allows the user to customize a web page by selecting a background color for the page, as shown in Example 3.13, as well as a color and font for the text. The following function shows you how to change font color and font family. Your page should offer the user more options than just the ones shown here.

```
<head>
<script type="text/javascript">
function customize()
{
     document.getElementById("p1").style.color="red";
     document.getElementById("p1").style.fontFamily="Arial";
}
</script>
</head>
<body>
<p id="p1">Hi there!</p>
<input type="button" onclick="customize()" value="customize" />
</body>
```

Save your page as `customize.html` and be sure to include an appropriate page title.

Case Studies

Greg's Gambits

Now you will add to Madame Vadoma's fortune telling abilities. On this new page Madame Vadoma will tell the player his or her fortune.

Open the `play_games.html` page and add a link, under the `Madame Vadoma Sees All!` link that links to the new page you will create. The new page title should be `Madame Vadoma Can Tell Your Fortune` and the filename should be `gregs_fortune2.html`.

Create this new page with the filename `gregs_fortune2.html`. You can use the page created earlier in the chapter as a template. The page title is `Madame Vadoma Can Tell Your Fortune` and this should also be the first header on the page.

In the content area, place a button that the player can click when he or she wants a fortune told.

Next, create at least 10 fortunes. You can create as many as you want but there should be at least 10. Here are a few sample fortunes to get you started:

- You will meet your soul mate very soon.
- You can get an A in this class if you work hard.
- A good job is in your future.
- No one likes a show-off.
- Don't believe everything you read.

Now create the JavaScript that will run when the player clicks the button. Use the `Math.random()` method to generate a random number between 1 and 10 (or 1 and however many fortunes you have created). Use a `switch` statement to display the fortune that corresponds to the random number that has been generated.

Test your page in at least two different browsers. Submit your work as instructed by your teacher.

Carla's Classroom

Now you will add to the addition exercise created earlier in this chapter. You will create pages with one or more of the following arithmetic exercises:

- Add two more levels of difficulty to the addition tests: adding floating point numbers (numbers with decimal parts) and adding fractions. Name this page `adv_addition.html`.
- Create three levels of multiplication problems: multiplying two single digit integers (from 1 to 10), multiplying two double-digit integers (from 1 to 99), and multiplying two double-digit floating point numbers. Name this page `multiply.html`.
- Create three levels of division problems using the three levels of difficulty as described for multiplication. Be sure to ensure that no division by zero error occurs. Name this page `divide.html`.

Each page you create should prompt the student to solve an arithmetic problem. There should be a minimum of five (preferably 10) problems at each level. Each problem should be created by generating two random numbers. The correct answer should be compared to the student's answer. If the answer is correct, a counter should be incremented. When the student has answered three problems correctly at any level, the program should progress to the next level. If a student goes through all the problems in one level without achieving three correct answers, the program should stop and a message should display telling the student that more practice is needed at that level. If the student successfully completes all levels, the message displayed should tell the student to move on to the next exercise.

Open the `math.html` file and add one or more links (depending on how many options you choose to program) under the `Addition Exercises`. The links should go to the new page or pages you create.

Use the `carla_adding.html` page that was created earlier in the chapter as a template to create these new arithmetic exercises for `Carla's Classroom`. In Chapter 4, we will learn how to generate as many problems as desired without repeating code. We will no longer be limited to five, six, ten, or any specific number of problems.

Test your page(s) in at least two different browsers. Be sure to test all possible combinations of both correct and incorrect responses at each level. Submit your work as instructed by your teacher.

Lee's Landscape

Add a page to the `Lee's Landscape` website that calculates the cost of various services offered. The page should display the services and their costs, as shown below. Then it will prompt the user to select a service along with one of the options shown and calculate the cost. If the user chooses to contract for the service for six months, there is a 10% discount. There is a 15% discount for a one-year contract and a 20% discount for a two-year contract.

Service	Options	
Lawn maintenance	Weekly: $15/service Twice a month: $25/service Monthly: $40/service	
Pest control	Monthly: $35/service Twice a year: $75/service Yearly: $150/service	
Tree and hedge trimming	Monthly: $25/service Twice a year: $75/service Yearly: $150/service	

The images used here are in the Student Data Files but you can use your own images too. Be sure to give this web page an appropriate page title,

such as `Lee's Landscape | Our Services`. Save this file with the filename `lee_service.html`. Add a link to the `Lee's Landscape` home page to this new page. Test your page in at least two different browsers. Be sure to test all possible combinations of each service and contract. Submit your work as instructed by your teacher.

Jackie's Jewelry

Add a page to the `Jackie's Jewelry` website that calculates shipping costs for items purchased according to the rates given below. The page should prompt the user to enter the amount of the purchase and the country where the items will be shipped. It should also allow free shipping if the user has a special promo code called `FREEJACKIE`.

- User has `FREEJACKIE` promo code: free
- Purchase is greater than or equal to $100 and user lives in the United States: free
- Purchase is greater than or equal to $150 and user lives in Canada or Mexico: free
- User lives in the United States:
  - Purchase between $50.00–$99.99: shipping is $10.00
  - Purchase between $20.00–$49.99: shipping is $8.00
  - Purchase less than $20.00: shipping is $5.00
- User lives in Canada or Mexico:
  - Purchase between $50.00–$99.99: shipping is $15.00
  - Purchase between $20.00–$49.99: shipping is $12.00
  - Purchase less than $20.00: shipping is $10.00
- User lives outside the United States, Canada, and Mexico:
  - Purchase between $50.00–$99.99: shipping is $25.00
  - Purchase between $20.00–$49.99: shipping is $20.00
  - Purchase less than $20.00: shipping is $15.00

Be sure to give this web page an appropriate page title, such as `Jackie's Jewelry | Shipping`. Save this file with the filename `jackie_shipping.html`. Add a link to the `Jackie's Jewelry` home page to this new page. Test your page in at least two different browsers. Be sure to test all possible combinations of purchases, addresses, and promo code. Submit your work as instructed by your teacher.