

## 1. (10 Points) Valid or Invalid syntax?

a. <pre>int numBooks = 3; int numNotebooks = 2; numBooks += numNotebooks;</pre>	<b>Valid</b>
b. <pre>System.out.print("Dog count = ", numDogs);</pre>	<b>Invalid</b>
c. <pre>int __numCars = 5;</pre>	<b>Valid</b>
d. <pre>String s1 = "Hello"; String s2 = "Goodbye"; s1 = s1 + " " + s2;</pre>	<b>Valid</b>
e. <pre>if (5 &lt; a &lt; 9) {     a = a * 5; } else {     a = a * 6; }</pre>	<b>Invalid</b>

2. (20 Points) During text messaging, people use abbreviations to save on typing. Expand the program below to recognize and translate the following abbreviations:

- LOL – Laughing Out Loud
- TMI – Too Much Information
- SMH – Shaking My Head

If the abbreviation is not included in the list, your program should output “Unknown Abbreviation” for the translation. **The input should be case insensitive** and the output should be as shown above.

```
import java.util.Scanner;

public class SMSAbbreviations1 {

    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter The SMS Abbreviation: ");
        String abbreviation = scnr.nextLine();

        String translation = "";

        /* Your solution goes here */
        if (abbreviation.equalsIgnoreCase("LOL")) {
            translation = "Laughing Out Loud";
        } else if (abbreviation.equalsIgnoreCase("TMI")) {
            translation = "Too Much Information";
        } else if (abbreviation.equalsIgnoreCase("SMH")) {
            translation = "Shaking My Head";
        } else {
            translation = "Unknown Abbreviation";
        }

        System.out.println(abbreviation + " means " + translation);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute  $x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ . You can assume that x, a, b and c are all **double** values. You must use **Math** operations where possible.

```
double bSquared = Math.pow(b, 2.0);  
x = (-b - Math.sqrt(bSquared - (4 * a * c))) / (2 * a);
```

4. (5 Points) Convert the binary number 01011010 to a decimal number.

5. (10 Points) Given the following code:

```
import java.util.Scanner;

public class Switch1 {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);
        System.out.println("Enter A Number From 1..6: ");
        int num = scnr.nextInt();

        switch (num) {
            case 1:
                System.out.println("One");
                break;
            case 2:
                System.out.println("Two");
            case 3:
                System.out.println("Three");
            case 4:
                System.out.println("Four");
                break;
            case 5:
                System.out.println("Five");
            case 6:
                System.out.println("Six");
                break;
            default:
                System.out.println("Invalid Number");
        }
    }
}
```

a. What is printed when the user enters 1?

One

b. What is printed when the user enters 3?

Three  
Four

c. What is printed when the user enters 5?

Five  
Six

d. What is printed when the user enters 6?

Six

e. What is printed when the user enters 8?

Invalid Number

6. (15 Points) Given the following string definition:

```
String str = "Java Programming Is Fun!";
```

a. Write the Java statement that would return the length of str.

```
str.length();
```

b. What is the length of str?

24

c. Write the Java statement to find the index of the substring "Is":

```
str.indexOf("Is");
```

d. What is the index of the substring "Fun"?

20

e. What does the Java statement `str.substring(5, 16)` return?

"Programming"

f. What is the Java statement to change the word "Fun" to the word "Incredible":

```
str.replace("Fun", "Incredible");
```

g. Write the Java statement to append to str the string " All the time!!!"

```
str = str.concat(" All the time!!!");
```

7. (20 Points) Write a **complete** Java program that prompts the user for *yearNumber*. Your program will then print out one of the following messages:

- *yearNumber* is a leap year
- *yearNumber* is not a leap year

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Year Number: ");
        int yearNumber = scnr.nextInt();

        boolean divisibleByFour      = (yearNumber % 4) == 0;
        boolean divisibleByHundred    = (yearNumber % 100) == 0;
        boolean divisibleByFourHundred = (yearNumber % 400) == 0;

        boolean leapYear;
        leapYear = (divisibleByFour && !divisibleByHundred) || divisibleByFourHundred;

        if (leapYear) {
            System.out.println(yearNumber + " is a leap year");
        } else {
            System.out.println(yearNumber + " is not a leap year");
        }

        return;
    }
}
```

8. (25 Points) Write a **complete** Java program that prompts the user for *monthNumber* (where 1 = January, 2 = February, ... , 12 = December) and *date* (a number from 1 .. 31). Your program will then print out one of the following messages:

- Error: *monthNumber* is not a valid month
- Error: *date* is not a valid date
- Error: *monthNumber* does not have *date* days
- *monthNumber date* is *monthName date*.

You can assume that February only has 28 days.

```
import java.util.Scanner;

public class MonthAndDate {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Month Number and a Date: ");
        int monthNumber = scnr.nextInt();
        int date = scnr.nextInt();

        String monthName = "";
        switch (monthNumber) {
            case 1:
                monthName = "January";
                break;
            case 2:
                monthName = "February";
                break;
            case 3:
                monthName = "March";
                break;
            case 4:
                monthName = "April";
                break;
            case 5:
                monthName = "May";
                break;
            case 6:
                monthName = "June";
                break;
            case 7:
                monthName = "July";
                break;
            case 8:
                monthName = "August";
                break;
            case 9:
                monthName = "September";
                break;
        }
    }
}
```

```
        case 10:
            monthName = "October";
            break;
        case 11:
            monthName = "November";
            break;
        case 12:
            monthName = "December";
            break;
        default:
            System.out.println("Error: " + monthNumber + " is not a valid month");
            return;
    }

    if ((date < 1) || date > 31) {
        System.out.println("Error: " + date + " is not a valid date");
        return;
    }

    if ((monthNumber == 2) && (date > 28)) {
        System.out.println("Error: " + monthNumber + " does not have " + date + " days");
        return;
    }

    if (date == 31) {
        switch (monthNumber) {
            case 4:
            case 6:
            case 9:
            case 11:
                System.out.println("Error: " + monthNumber + " does not have "
                    + date + " days");
                return;
        }
    }

    System.out.println(monthNumber + " " + date + " is " + monthName + " " + date);

    return;
}
}
```

## 1. (10 Points) Valid or invalid syntax?

a. <pre>int numBooks = 3; int numNotebooks = 2; numBooks += numNotebooks;</pre>	<b>Invalid</b>
b. <pre>system.out.print("Dog count = " + numDogs);</pre>	<b>Invalid</b>
c. <pre>int 3rdPlace = 6;</pre>	<b>Invalid</b>
d. <pre>String s1 = "Hello"; String s2 = "Goodbye"; s1 = s1.concat(s1);</pre>	<b>Valid</b>
e. <pre>if ((a &gt; 5) &amp;&amp; (a &lt; 9))     a = a * 7; else     a = a * 3;</pre>	<b>Valid</b>

2. (20 Points) During text messaging, people use abbreviations to save on typing. Expand the program below to recognize and translate the following abbreviations:

- BFF – Best Friend Forever
- IMHO – In My Humble Opinion
- 2QT – Too Cute

If the abbreviation is not included in the list, your program should output “Unknown Abbreviation” for the translation. **The input should be case insensitive** and the output should be as shown above.

```
import java.util.Scanner;

public class SMSAbbreviations2 {

    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter The SMS Abbreviation: ");
        String abbreviation = scnr.nextLine();

        String translation = "";

        /* Your solution goes here */
        if (abbreviation.equalsIgnoreCase("BFF")) {
            translation = "Best Friend Forever";
        } else if (abbreviation.equalsIgnoreCase("IMHO")) {
            translation = "In My Humble Opinion";
        } else if (abbreviation.equalsIgnoreCase("2QT")) {
            translation = "Too Cute";
        } else {
            translation = "Unknown Abbreviation";
        }

        System.out.println(abbreviation + " means " + translation);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute  $x = \frac{-c - \sqrt{a^3 - 5bc}}{4b}$ . You can assume that x, a, b and c are all **double** values. You must use `Math` operations where possible.

```
double aCubed = Math.pow(a, 3.0);  
x = (-c - Math.sqrt(aCubed - (5 * b * c))) / (4 * b);
```

4. (5 Points) Convert the binary number 11001101 to a decimal number.

**205**

5. (10 Points) Given the following code:

```
import java.util.Scanner;

public class Switch2 {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);
        System.out.println("Enter A Number From 1..6: ");
        int num = scnr.nextInt();

        switch (num) {
            case 1:
                System.out.println("One");
            case 2:
                System.out.println("Two");
                break;
            case 3:
                System.out.println("Three");
            case 4:
                System.out.println("Four");
                break;
            case 5:
                System.out.println("Five");
            case 6:
                System.out.println("Six");
                break;
            default:
                System.out.println("Invalid Number");
        }
    }
}
```

a. What is printed when the user enters 1?

One  
Two

d. What is printed when the user enters 5?

Five  
Six

b. What is printed when the user enters 3?

Three  
Four

e. What is printed when the user enters 7?

Invalid Number

c. What is printed when the user enters 4?

Four

6. (15 Points) Given the following string definition:

```
String str = "CMP-167 Is The Best Class!";
```

a. Write the Java statement that would return the length of str.

```
str.length();
```

b. What is the length of str?

26

c. Write the Java statement to find the index of the substring "Best":

```
str.indexOf("Best");
```

d. What is the index of the substring "Class"?

20

e. What does the Java statement `str.substring(15, 25)` return?

```
"Best Class"
```

f. What is the Java statement to change the word "Best" to the word "Most Amazing":

```
str.replace("Best", "Most Amazing");
```

g. Write the Java statement to append to str the string " Ever!!!"

```
str = str.concat(" Ever!!!");
```

7. (20 Points) Write a **complete** Java program that prompts the user for *yearNumber*. Your program will then print out one of the following messages:

- *yearNumber* is a leap year
- *yearNumber* is not a leap year

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Year Number: ");
        int yearNumber = scnr.nextInt();

        boolean divisibleByFour      = (yearNumber % 4) == 0;
        boolean divisibleByHundred    = (yearNumber % 100) == 0;
        boolean divisibleByFourHundred = (yearNumber % 400) == 0;

        boolean leapYear;
        leapYear = (divisibleByFour && !divisibleByHundred) || divisibleByFourHundred;

        if (leapYear) {
            System.out.println(yearNumber + " is a leap year");
        } else {
            System.out.println(yearNumber + " is not a leap year");
        }

        return;
    }
}
```

8. (25 Points) Write a **complete** Java program that prompts the user for *monthNumber* (where 1 = January, 2 = February, ... , 12 = December) and *date* (a number from 1 .. 31). Your program will then print out one of the following messages:

- Error: *monthNumber* is not a valid month
- Error: *date* is not a valid date
- Error: *monthNumber* does not have *date* days
- *monthNumber date* is *monthName date*.

You can assume that February only has 28 days.

```
import java.util.Scanner;

public class MonthAndDate {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Month Number and a Date: ");
        int monthNumber = scnr.nextInt();
        int date = scnr.nextInt();

        String monthName = "";
        switch (monthNumber) {
            case 1:
                monthName = "January";
                break;
            case 2:
                monthName = "February";
                break;
            case 3:
                monthName = "March";
                break;
            case 4:
                monthName = "April";
                break;
            case 5:
                monthName = "May";
                break;
            case 6:
                monthName = "June";
                break;
            case 7:
                monthName = "July";
                break;
            case 8:
                monthName = "August";
                break;
            case 9:
                monthName = "September";
                break;
```

```
        case 10:
            monthName = "October";
            break;
        case 11:
            monthName = "November";
            break;
        case 12:
            monthName = "December";
            break;
        default:
            System.out.println("Error: " + monthNumber + " is not a valid month");
            return;
    }

    if ((date < 1) || date > 31) {
        System.out.println("Error: " + date + " is not a valid date");
        return;
    }

    if ((monthNumber == 2) && (date > 28)) {
        System.out.println("Error: " + monthNumber + " does not have " + date + " days");
        return;
    }

    if (date == 31) {
        switch (monthNumber) {
            case 4:
            case 6:
            case 9:
            case 11:
                System.out.println("Error: " + monthNumber + " does not have "
                    + date + " days");
                return;
        }
    }

    System.out.println(monthNumber + " " + date + " is " + monthName + " " + date);

    return;
}
}
```

## 1. (10 Points) Valid or invalid syntax?

a. <pre>int numBooks = 3; int numNotebooks = 2; numBooks -= numNotebooks;</pre>	<b>Valid</b>
b. <pre>System.out.print("Dog count = " numDogs);</pre>	<b>Invalid</b>
c. <pre>int very short = 6;</pre>	<b>Invalid</b>
d. <pre>String s1 = "Hello"; String s2 = "Goodbye"; boolean b = s1 &gt; s2;</pre>	<b>Invalid</b>
e. <pre>if ((a =&gt; 5) &amp;&amp; (a =&lt; 9))     a = a * 7; else     a = a * 3;</pre>	<b>Invalid</b>

2. (20 Points) During text messaging, people use abbreviations to save on typing. Expand the program below to recognize and translate the following abbreviations:

- AAS – Alive And Smiling
- AYK – As You Know
- BAU – Business As Usual

If the abbreviation is not included in the list, your program should output “Unknown Abbreviation” for the translation. **The input should be case insensitive** and the output should be as shown above.

```
import java.util.Scanner;

public class SMSAbbreviations3 {

    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter The SMS Abbreviation: ");
        String abbreviation = scnr.nextLine();

        String translation = "";

        /* Your solution goes here */
        if (abbreviation.equalsIgnoreCase("AAS")) {
            translation = "Alive And Smiling";
        } else if (abbreviation.equalsIgnoreCase("AYK")) {
            translation = "As You Know";
        } else if (abbreviation.equalsIgnoreCase("BAU")) {
            translation = "Business As Usual";
        } else {
            translation = "Unknown Abbreviation";
        }

        System.out.println(abbreviation + " means " + translation);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute  $x = \frac{-b + \sqrt{b^3 - 4ac}}{2a}$ . You can assume that x, y and z are all **double** values. You must use `Math` operations where possible.

```
double bCubed = Math.pow(b, 3.0);  
x = (-b + Math.sqrt(bCubed - (4 * a * c))) / (2 * a);
```

4. (5 Points) Convert the binary number 01110110 to a decimal number.

118

5. (10 Points) Given the following code:

```
import java.util.Scanner;

public class Switch3 {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);
        System.out.println("Enter A Number From 1..6: ");
        int num = scnr.nextInt();

        switch (num) {
            case 1:
                System.out.println("One");
                break;
            case 2:
                System.out.println("Two");
            case 3:
                System.out.println("Three");
                break;
            case 4:
                System.out.println("Four");
            case 5:
                System.out.println("Five");
                break;
            case 6:
                System.out.println("Six");
            default:
                System.out.println("Invalid Number");
        }
    }
}
```

a. What is printed when the user enters 1?

One

b. What is printed when the user enters 2?

Two  
Three

c. What is printed when the user enters 4?

Four  
Five

d. What is printed when the user enters 6?

Six  
Invalid Number

e. What is printed when the user enters 9?

Invalid Number

6. (15 Points) Given the following string definition:

```
String str = "Peace And Education For All!";
```

a. Write the Java statement that would return the length of str.

```
str.length();
```

b. What is the length of str?

28

c. Write the Java statement to find the index of the substring "Education":

```
str.indexOf("Education");
```

d. What is the index of the substring "For"?

20

e. What does the Java statement `str.substring(10, 19)` return?

```
"Education"
```

f. What is the Java statement to change the word " And " to the word ", Justice And ":

```
str.replace(" And ", ", Justice And ");
```

g. Write the Java statement to append to str the string " Now!!!"

```
str = str.concat(" Now!!!");
```

7. (20 Points) Write a **complete** Java program that prompts the user for *yearNumber*. Your program will then print out one of the following messages:

- *yearNumber* is a leap year
- *yearNumber* is not a leap year

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Year Number: ");
        int yearNumber = scnr.nextInt();

        boolean divisibleByFour      = (yearNumber % 4) == 0;
        boolean divisibleByHundred    = (yearNumber % 100) == 0;
        boolean divisibleByFourHundred = (yearNumber % 400) == 0;

        boolean leapYear;
        leapYear = (divisibleByFour && !divisibleByHundred) || divisibleByFourHundred;

        if (leapYear) {
            System.out.println(yearNumber + " is a leap year");
        } else {
            System.out.println(yearNumber + " is not a leap year");
        }

        return;
    }
}
```

8. (25 Points) Write a **complete** Java program that prompts the user for *monthNumber* (where 1 = January, 2 = February, ... , 12 = December) and *date* (a number from 1 .. 31). Your program will then print out one of the following messages:

- Error: *monthNumber* is not a valid month
- Error: *date* is not a valid date
- Error: *monthNumber* does not have *date* days
- *monthNumber date* is *monthName date*.

You can assume that February only has 28 days.

```
import java.util.Scanner;

public class MonthAndDate {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Month Number and a Date: ");
        int monthNumber = scnr.nextInt();
        int date = scnr.nextInt();

        String monthName = "";
        switch (monthNumber) {
            case 1:
                monthName = "January";
                break;
            case 2:
                monthName = "February";
                break;
            case 3:
                monthName = "March";
                break;
            case 4:
                monthName = "April";
                break;
            case 5:
                monthName = "May";
                break;
            case 6:
                monthName = "June";
                break;
            case 7:
                monthName = "July";
                break;
            case 8:
                monthName = "August";
                break;
            case 9:
                monthName = "September";
                break;
        }
    }
}
```

```
        case 10:
            monthName = "October";
            break;
        case 11:
            monthName = "November";
            break;
        case 12:
            monthName = "December";
            break;
        default:
            System.out.println("Error: " + monthNumber + " is not a valid month");
            return;
    }

    if ((date < 1) || date > 31) {
        System.out.println("Error: " + date + " is not a valid date");
        return;
    }

    if ((monthNumber == 2) && (date > 28)) {
        System.out.println("Error: " + monthNumber + " does not have " + date + " days");
        return;
    }

    if (date == 31) {
        switch (monthNumber) {
            case 4:
            case 6:
            case 9:
            case 11:
                System.out.println("Error: " + monthNumber + " does not have "
                    + date + " days");
                return;
        }
    }

    System.out.println(monthNumber + " " + date + " is " + monthName + " " + date);

    return;
}
}
```

## 1. (10 Points) Valid or invalid syntax?

a. <pre>int numBooks = 3; int numNotebooks = 2; numBooks -= numNotebooks;</pre>	<b>Invalid</b>
b. <pre>System.out.print("Dog count = " + numDogs);</pre>	<b>Valid</b>
c. <pre>int veryTall! = 7;</pre>	<b>Invalid</b>
d. <pre>String s1 = "Hello"; String s2 = "Goodbye"; int i = s1.poaitionOf('H');</pre>	<b>Invalid</b>
e. <pre>if ((a &gt;= 5)    (a &lt;= 9))     a = a * 7; else     a = a * 3;</pre>	<b>Valid</b>

2. (20 Points) During text messaging, people use abbreviations to save on typing. Expand the program below to recognize and translate the following abbreviations:

- BRB – Be Right Back
- IDST – I Did Not Say That
- NMP – Not My Problem

If the abbreviation is not included in the list, your program should output “Unknown Abbreviation” for the translation. **The input should be case insensitive** and the output should be as shown above.

```
import java.util.Scanner;

public class SMSAbbreviations4 {

    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter The SMS Abbreviation: ");
        String abbreviation = scnr.nextLine();

        String translation = "";

        /* Your solution goes here */
        if (abbreviation.equalsIgnoreCase("BRB")) {
            translation = "Be Right Back";
        } else if (abbreviation.equalsIgnoreCase("IDST")) {
            translation = "I did Not Say That";
        } else if (abbreviation.equalsIgnoreCase("NMP")) {
            translation = "Not My Problem";
        } else {
            translation = "Unknown Abbreviation";
        }

        System.out.println(abbreviation + " means " + translation);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute  $x = \frac{-a - \sqrt{c^3 - 7ab}}{4c}$ . You can assume that x, a, b and c are all **double** values. You must use **Math** operations where possible.

```
double cCubed = Math.pow(c, 3.0);  
x = (-a - Math.sqrt(cCubed - (7 * a * b))) / (4 * c);
```

4. (5 Points) Convert the binary number 01101110 to a decimal number.

110

5. (10 Points) Given the following code:

```
import java.util.Scanner;

public class Switch4 {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);
        System.out.println("Enter A Number From 1..6: ");
        int num = scnr.nextInt();

        switch (num) {
            case 1:
                System.out.println("One");
            case 2:
                System.out.println("Two");
                break;
            case 3:
                System.out.println("Three");
            case 4:
                System.out.println("Four");
                break;
            case 5:
                System.out.println("Five");
            case 6:
                System.out.println("Six");
            default:
                System.out.println("Invalid Number");
        }
    }
}
```

a. What is printed when the user enters 1?

One  
Two

b. What is printed when the user enters 2?

Two

c. What is printed when the user enters 3?

Three  
Four

d. What is printed when the user enters 5?

Five  
Six  
Invalid Number

e. What is printed when the user enters 8?

Invalid Number

6. (15 Points) Given the following string definition:

```
String str = "Cold Weather Is Terrible!";
```

a. Write the Java statement that would return the length of str.

```
str.length();
```

b. What is the length of str?

25

c. Write the Java statement to find the index of the substring "Is":

```
str.indexOf("Is");
```

d. What is the index of the substring "Terrible"?

16

e. What does the Java statement `str.substring(5, 15)` return?

```
"Weather Is"
```

f. What is the Java statement to change the word "Terrible" to the word "The Worst":

```
str.replace("Terrible", "The Worst");
```

g. Write the Java statement to append to str the string " Especially In Winter!!!"

```
str = str.concat(" Especially In Winter!!!");
```

7. (20 Points) Write a **complete** Java program that prompts the user for *yearNumber*. Your program will then print out one of the following messages:

- *yearNumber* is a leap year
- *yearNumber* is not a leap year

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Year Number: ");
        int yearNumber = scnr.nextInt();

        boolean divisibleByFour      = (yearNumber % 4) == 0;
        boolean divisibleByHundred    = (yearNumber % 100) == 0;
        boolean divisibleByFourHundred = (yearNumber % 400) == 0;

        boolean leapYear;
        leapYear = (divisibleByFour && !divisibleByHundred) || divisibleByFourHundred;

        if (leapYear) {
            System.out.println(yearNumber + " is a leap year");
        } else {
            System.out.println(yearNumber + " is not a leap year");
        }

        return;
    }
}
```

8. (25 Points) Write a **complete** Java program that prompts the user for *monthNumber* (where 1 = January, 2 = February, ... , 12 = December) and *date* (a number from 1 .. 31). Your program will then print out one of the following messages:

- Error: *monthNumber* is not a valid month
- Error: *date* is not a valid date
- Error: *monthNumber* does not have *date* days
- *monthNumber date* is *monthName date*.

You can assume that February only has 28 days.

```
import java.util.Scanner;

public class MonthAndDate {
    public static void main(String[] args) {
        Scanner scnr = new Scanner(System.in);

        System.out.println("Enter A Month Number and a Date: ");
        int monthNumber = scnr.nextInt();
        int date = scnr.nextInt();

        String monthName = "";
        switch (monthNumber) {
            case 1:
                monthName = "January";
                break;
            case 2:
                monthName = "February";
                break;
            case 3:
                monthName = "March";
                break;
            case 4:
                monthName = "April";
                break;
            case 5:
                monthName = "May";
                break;
            case 6:
                monthName = "June";
                break;
            case 7:
                monthName = "July";
                break;
            case 8:
                monthName = "August";
                break;
            case 9:
                monthName = "September";
                break;
```

```
        case 10:
            monthName = "October";
            break;
        case 11:
            monthName = "November";
            break;
        case 12:
            monthName = "December";
            break;
        default:
            System.out.println("Error: " + monthNumber + " is not a valid month");
            return;
    }

    if ((date < 1) || date > 31) {
        System.out.println("Error: " + date + " is not a valid date");
        return;
    }

    if ((monthNumber == 2) && (date > 28)) {
        System.out.println("Error: " + monthNumber + " does not have " + date + " days");
        return;
    }

    if (date == 31) {
        switch (monthNumber) {
            case 4:
            case 6:
            case 9:
            case 11:
                System.out.println("Error: " + monthNumber + " does not have "
                    + date + " days");
                return;
        }
    }

    System.out.println(monthNumber + " " + date + " is " + monthName + " " + date);

    return;
}
}
```