

1. (10 Points) Valid or Invalid syntax?

a. <pre>/* This is a block Comment that Spans 3 lines */</pre>	Valid
b. <pre>System.out.print(numDogs).</pre>	Invalid
c. <pre>int numCars = 5;</pre>	Valid
d. <pre>if(i == 5) i += 1; k -= 1; else i -= 1; k += 1;</pre>	Invalid
e. <pre>if(a > 5 && < 9) { a = a * 5; } else { a = a * 6; }</pre>	Invalid

2. (20 Points) A cashier distributes change using the maximum number of ten-dollar bills, followed by the maximum number of five-dollar bills, followed by one-dollar bills.

Add the statements to compute numTens, numFives and numOnes, given amountToChange. Hint: The / and % operators are useful.

```
import java.util.Scanner;

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

        System.out.println("Enter The Amount To Change: ");
        int amountToChange = scnr.nextInt();

        int numTens = 0;
        int numFives = 0;
        int numOnes = 0;

        /* Your solution goes here */

        numTens = amountToChange / 10;

        amountToChange = amountToChange % 10;

        numFives = amountToChange / 5;

        numOnes = amountToChange % 5;

        System.out.println("numTens : " + numTens);
        System.out.println("numFives: " + numFives);
        System.out.println("numOnes : " + numOnes);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute $x = \sqrt{y^2 + z^2}$. You can assume that x, y and z are all **double** values.

```
double ySquared = Math.pow(y, 2.0);  
double zSquared = Math.pow(z, 2.0);  
x = Math.sqrt(ySquared + zSquared);
```

4. (5 Points) Convert the binary number 00101010 to a decimal number.

42

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..4: ");
        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");
            default:
                System.out.println("Invalid Number");
        }
        return;
    }
}
```

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 3?

Three

d. What is printed when the user enters 4?

Four
Invalid Number

e. What is printed when the user enters 5?

Invalid Number

6. (12 Points) Given the following string definition:

```
String str = "I like the Easter bunny";
```

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

```
len = str.length();
```

- b. What is the length of str?

23

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

```
i = str.indexOf("Easter");
```

- d. What is the index of the substring "bunny"?

18

- e. What is the Java statement to change the word "like" to the word "love":

```
str = str.replaceAll("like", "love");
```

- f. Write the Java statement to append " all the time!!!"

```
str = str + " 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. (28 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>// This is a block Comment that Spans 3 lines //</pre>	Invalid
b. <pre>System.out.print("Dogs: " numDogs);</pre>	Invalid
c. <pre>int tall = 6;</pre>	Valid
d. <pre>if(i==5) { i += 1; k -= 1; } else { i -= 1; k += 1;</pre>	Invalid
e. <pre>if((a > 5) && (a < 9)) { a = a * 7; } else { a = a * 3; }</pre>	Valid

2. (20 Points) A cashier distributes change using the maximum number of twenty-dollar bills, followed by the maximum number of ten-dollar bills, followed by five-dollar bills. You can assume that there will not be any one-dollar bills in the change.

Add the statements to compute numTwenties, numTens and numFives, given amountToChange. Hint: The / and % operators are useful.

```
import java.util.Scanner;

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

        System.out.println("Enter The Amount To Change: ");
        int amountToChange = scnr.nextInt();

        int numTwenties = 0;
        int numTens = 0;
        int numFives = 0;

        /* Your solution goes here */

        numTwenties = amountToChange / 20;

        amountToChange = amountToChange % 20;

        numTens = amountToChange / 10;

        amountToChange = amountToChange % 10;

        numFives = amountToChange / 5;

        System.out.println("numTwenties : " + numTwenties);
        System.out.println("numTens      : " + numTens);
        System.out.println("numFives   : " + numFives);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute $x = \sqrt{y^3 - z^3}$. You can assume that x, y and z are all **double** values.

```
double yCubed = Math.pow(y, 3.0);  
double zCubed = Math.pow(z, 3.0);  
x = Math.sqrt(yCubed - zCubed);
```

4. (5 Points) Convert the binary number 10010101 to a decimal number.

149

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..4: ");
        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;
            default:
                System.out.println("Invalid Number");
        }

        return;
    }
}
```

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 4?

Four

e. What is printed when the user enters 5?

Invalid Number

6. (12 Points) Given the following string definition:

```
String str = "March is the month of spring madness";
```

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

```
len = str.length();
```

- b. What is the length of str?

36

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

```
i = str.indexOf("Spring");
```

- d. What is the index of the substring "madness"?

29

- e. What is the Java statement to change the word "month of" to the word "time for":

```
str = str.replaceAll("month of", "time for");
```

- f. Write the Java statement to append " all the time!!!"

```
str = str + " every year!!!";
```

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. (28 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>/* * This is a block * Comment that * Spans 3 lines */</pre>	Valid
b. <pre>System.out.print("Amy // Michael");</pre>	Valid
c. <pre>int short = 6;</pre>	Invalid
d. <pre>if(i == 5) { i += 1; k -= 1; } else { i -= 1; k += 1; }</pre>	Valid
e. <pre>if(a < 7 > 9) { a = a * 2; } else { a = a * 3; }</pre>	Invalid

2. (20 Points) A cashier distributes change using the maximum number of ten-dollar bills, followed by the maximum number of five-dollar bills, followed by one-dollar bills.

Add the statements to compute numTens, numFives and numOnes, given amountToChange. Hint: The / and % operators are useful.

```
import java.util.Scanner;

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

        System.out.println("Enter The Amount To Change: ");
        int amountToChange = scnr.nextInt();

        int numTens = 0;
        int numFives = 0;
        int numOnes = 0;

        /* Your solution goes here */

        numTens = amountToChange / 10;

        amountToChange = amountToChange % 10;

        numFives = amountToChange / 5;

        numOnes = amountToChange % 5;

        System.out.println("numTens : " + numTens);
        System.out.println("numFives: " + numFives);
        System.out.println("numOnes : " + numOnes);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute $x = \sqrt{(\tan y)^2 - (\sin z)^2}$. You can assume that x, y and z are all **double** values.

```
double tanYSquared = Math.pow(Math.tan(y), 2.0);  
double sinZSquared = Math.pow(Math.sin(z), 2.0);  
x = Math.sqrt(tanYSquared - sinZSquared);
```

4. (5 Points) Convert the decimal number 79 to an 8-bit binary number.

01001111

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..4: ");
        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");
            default:
                System.out.println("Invalid Number");
        }
        return;
    }
}
```

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 3?

Three

d. What is printed when the user enters 4?

Four
Invalid Number

e. What is printed when the user enters 5?

Invalid Number

6. (12 Points) Given the following string definition:

```
String str = "This winter was not very cold";
```

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

```
len = str.length();
```

- b. What is the length of str?

29

- c. Write the Java statement to find the index of the substring "not":

```
i = str.indexOf("not");
```

- d. What is the index of the substring "cold"?

25

- e. What is the Java statement to change the word "winter" to the word "March":

```
str = str.replaceAll("winter", "March");
```

- f. Write the Java statement to append ", that is great!!!"

```
str = str + ", that is great!!!";
```

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. (28 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>/* This is a block // line 1 Comment that // line 2 Spans 3 lines // line 3 */</pre>	Valid
b. <pre>System.print(numDogs);</pre>	Invalid
c. <pre>int very tall = 7;</pre>	Invalid
d. <pre>if(i == 5) i += 1; else i -= 1;</pre>	Valid
e. <pre>if((a < 7) a > 9) { a = a * 7; } else { a = a * 4; }</pre>	Invalid

2. (20 Points) A cashier distributes change using the maximum number of twenty-dollar bills, followed by the maximum number of ten-dollar bills, followed by five-dollar bills. You can assume that there will not be any one-dollar bills in the change.

Add the statements to compute numTwenties, numTens and numFives, given amountToChange.
Hint: The / and % operators are useful.

```
import java.util.Scanner;

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

        System.out.println("Enter The Amount To Change: ");
        int amountToChange = scnr.nextInt();

        int numTwenties = 0;
        int numTens = 0;
        int numFives = 0;

        /* Your solution goes here */

        numTwenties = amountToChange / 20;

        amountToChange = amountToChange % 20;

        numTens = amountToChange / 10;

        amountToChange = amountToChange % 10;

        numFives = amountToChange / 5;

        System.out.println("numTwenties : " + numTwenties);
        System.out.println("numTens      : " + numTens);
        System.out.println("numFives   : " + numFives);

        return;
    }
}
```

3. (10 Points) Write the Java statements to compute $x = \sqrt{(\cos y)^3 + (\tan z)^3}$. You can assume that x, y and z are all **double** values.

```
double cosYCubed = Math.pow(Math.cos(y), 3.0);  
double tanZCubed = Math.pow(Math.tan(z), 3.0);  
x = Math.sqrt(cosYCubed + tanZCubed);
```

4. (5 Points) Convert the decimal number 59 to an 8-bit binary number.

```
00111011
```

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..4: ");
        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;
            default:
                System.out.println("Invalid Number");
        }

        return;
    }
}
```

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 4?

Four

e. What is printed when the user enters 5?

Invalid Number

6. (12 Points) Given the following string definition:

```
String str = "I like programming in Java";
```

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

```
len = str.length();
```

- b. What is the length of str?

26

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

```
i = str.indexOf("in");
```

- d. What is the index of the substring "Java"?

22

- e. What is the Java statement to change the word "like" to the word "love":

```
str = str.replaceAll("like", "love");
```

- f. Write the Java statement to append ", it is fun!!!"

```
str = str + ", it is fun!!!";
```

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. (28 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;
}
}
```