1 Write a declaration for a String variable called `change` and initialize it to the characters stored in another String object called `original` with all 'e' characters changed to 'j'.

String original = "beneneNecte"; *//String object called 'original' must exist*
String change = original.replace('e', 'j'); *//change is original with e's changed to j's*

2 Write code to declare and instantiate an object of the `Random` class (call the object reference variable `rand`). Then write a list of expressions using the `nextInt` method that generates random numbers in the following specified ranges, including the endpoints. Use the version of the `nextInt` method that accepts a single integer parameter.

f. `-10 to 15`

```
import java.util.Random;  //Random class is needed

public class Randnumber2
{
	//-------------------------------------------------------
	//Write code to declare and instantiate an object of
	//the Random class (call the object reference variable rand).
	//Then write a list of expressions using the nextInt method
	//that generates random numbers in the following specified
	//ranges, including the endpoints. Use the version of the
	//nextInt method that accepts a single integer parameter.
	//-------------------------------------------------------

	public static void main(String[] args)
	{
		Random rand = new Random();
		//object reference variable rand
		int numb1;
		//some place-holder


		numb1 = rand.nextInt(26) - 10;
		//Spec said -10 to 15 (including the endpoints)

		System.out.println("Output should be from -10 to 15 => " +
					numb1); //Print out

	}


}
```

3 Write code statements that prompt for and read a double value from the user, and then print the result of raising that value to the fourth power. Output the results to three decimal places.

```
import java.util.Scanner;       //Needed this class for input
import java.text.DecimalFormat; //Needed to format numbers to pattern


public class Part3
{
	public static void main(String[] args)
	{
		double numb1, numb2; //numb1 is raw data, numb2 is (numb1^4)


		Scanner scan = new Scanner(System.in); //Create scan object
		                                       //to get input

		System.out.print("Enter a double precision number: "); //Prompt
		numb1 = scan.nextDouble();//Get the keyboard input of double type

		numb2 = Math.pow(numb1, 4); //Raised to the 4th power and stored
		//in numb2

		DecimalFormat fmt1 = new DecimalFormat("0.###");
		//fmt1 is an instance of DecimalFormat class
		//fmt1 created with pattern for 3 decimal places.

		System.out.println("The result is: " + fmt1.format(numb2));
		//Prints out the formatted result
	}


}
```

4 Write a declaration for an enumerated type that represents the days of the week

enum DaysOfTheWeek {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday} //enum Identifier {item1, item2,...,itemN}

4.3 What is wrong with the following code fragment? Will this code compile if it is part of an otherwise valid program? Explain.

```
if (length = MIN_LENGTH)

System.out.println ("The length is minimal.");
```

Here is the compilation error should the code above be placed in a valid program

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
	Type mismatch: cannot convert from int to boolean

	at Generic.main(Generic.java:40)
```

The conditional statement inside the if() must evaluate to a boolean result. Which it doesn't at the moment. To correct this problem, change the code to:

```
if (length == MIN_LENGTH)
System.out.println ("The length is minimal.");
```

5 Put the following list of strings in lexicographic order as if determined by the compareTo method of the String class. Consult the Unicode chart in Appendix C in the book.

```
    "fred"              \u(0066, 0072, 0065, 0064)
    "Ethel"             \u(0045, 0074, 0068, 0065, 006c)
    "?-?-?-?"           \u(003f, 002d, 003f, 002d, 003f, 002d, 003f)
    "{([])}"            \u(007b, 0028, 005b, 005d, 0029, 007d)
    "Lucy"              \u(004c, 0075, 0063, 0079)
    "ricky"             \u(0072, 0069, 0063, 006b, 0079)
    "book"              \u(0062, 006f, 006f, 006b)
    "******"            \u(002a, 002a, 002a, 002a, 002a, 002a)
    "12345"             \u(0031, 0032, 0033, 0034, 0035)
    " "                 \u0020
    "HEPHALUMP"   \u(0048,0045,0050,0048,0041,004c,0055,004d,0050)
    "bookkeeper”
    \u(0062,006f,006f,006b,006b,0065,0065,0070,0065,0072)

    "6789"        \u(0036,0037,0038,0039)
    ";+<?"        \u(003b,002b,003c,003f)
    "
    ^^^^^^^^^^
    "
    \u(000a,005e,005e,005e,005e,005e,005e,005e,005e,005e,005e,000a)
"hephalump"
\u(0068,0065,0070,0068,0061,006c,0075,006d,0070)
```

I do not understand this question. compareTo() takes existing string and another string to compare lexicographically the Unicode values of the 2 strings. I do not quite know what to do here.

6 Transform the while loop given below into an equivalent for loop. (Make sure it produces the same output.)

```
int num = 1;
while (num < 20)
{
	num++;
	System.out.println (num);
}
```

Changed to:

```
int num = 1;
for(++num; num <= 20; num++)
{
	System.out.println (num);
}
```

7 Write an application that creates and prints a random phone number of the form XXX-XXX-XXXX. Include the dashes in the output. Do not let the first three digits contain an 8 or 9 (but don't be more restrictive than that), and make sure that the second set of three digits is not greater than 742. Hint: Think through the easiest way to construct the phone number. Each digit does not have to be determined separately.

```
import java.util.Random;  //Random class is needed

public class PhoneNumber
{
	//-------------------------------------------------------
	///Write an application that creates and prints a random
	//phone number of the form XXX-XXX-XXXX. Include the dashes
	//in the output. Do not let the first three digits contain
	//an 8 or 9 (but don't be more restrictive than that), and
	//make sure that the second set of three digits is not greater
	//than 742. Hint: Think through the easiest way to construct
	//the phone number. Each digit does not have to be determined
	//separately.
	//-------------------------------------------------------
```

```
	public static void main(String[] args)
	{
		Random rand = new Random(); //object reference variable rand

		int numb1 = 0,numb2 = 0,numb3 = 0; //first 3 digits
		int numb4 = 0,numb5 = 0,numb6 = 0; //second group of 3 digits
		int numb7 = 0,numb8 = 0,numb9 = 0,numb10 = 0;//last group of 4
		//digits
		int groupsum = 742; //Preset groupsum for while loop

		//Generate the first 3 digits separately
		//Each digit goes from 1 to 7 to exclude 8 or 9
		numb1 = rand.nextInt(7) + 1; //First digit
		numb2 = rand.nextInt(7) + 1; //Second digit
		numb3 = rand.nextInt(7) + 1; //Third digit

		//Generate the second 3 digits separately
		//Each digit goes from 1 to 9
		//The sum must not exceed 742
		//Right now groupsum is 742. Loop
		//until this condition becomes false
		while(groupsum >= 742)
		{
			numb4 = rand.nextInt(10); //Fourth digit
			numb5 = rand.nextInt(10); //Fifth digit
			numb6 = rand.nextInt(10); //Sixth digit

			groupsum = ((numb4 * 100) + (numb5 * 10) + numb6);
		}

		//Generate the last 4 digits separately
		//Each digit goes from 0 to 9 inclusively
		numb7 = rand.nextInt(10); //Seventh digit
		numb8 = rand.nextInt(10); //Eighth digit
		numb9 = rand.nextInt(10); //Ninth digit
		numb10 = rand.nextInt(10); //Tenth digit

		System.out.println("The 10-digit phone number is => " + numb1 +
		numb2 + numb3 + "-" + numb4 + numb5 + numb6 + "-"
		+ numb7 + numb8 + numb9 + numb10);//Print out

	}


}
```

8 Write an application that reads the (x,y) coordinates for two points. Compute the distance between the two points using the following formula:

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
import java.util.Scanner;        //Needed this class for input
import java.text.DecimalFormat; //Needed to format numbers to pattern

public class Distance
{
	//-------------------------------------------------------
	//Write an application that reads the (x,y) coordinates for
	//two points. Compute the distance between the two points
	//using the following formula:
	//
	//          Distance = srqt((x2 - x1)^2 + (y2 - y1)^2)
	//
	//-------------------------------------------------------

	public static void main(String[] args)
	{
		double x1=0.0,x2=0.0,y1=0.0,y2=0.0,distance=0.0;
		//variables for calculating distance of 2 points

		DecimalFormat fmt1 = new DecimalFormat("0.###");

		Scanner scan = new Scanner(System.in);//Keyboard input

		System.out.print("Enter x1: ");
		x1 = scan.nextDouble();
		System.out.print("Enter y1: ");
		y1 = scan.nextDouble();

		System.out.print("Enter x2: ");
		x2 = scan.nextDouble();
		System.out.print("Enter y2: ");
		y2 = scan.nextDouble();


		distance = Math.sqrt((Math.pow((x2 - x1), 2)) + (Math.pow((y2 -
		y1), 2)));
		//      Distance = srqt((x2 - x1)^2 + (y2 - y1)^2)
		//      Assuming serious data entries

		System.out.println("The distance is: " + fmt1.format(distance) +
		" units");

	}
}
```

9 Write an application that reads the lengths of the sides of a triangle from the user. Compute the area of the triangle using Heron's formula (below), in which s represents half of the perimeter of the triangle, and a,b, and c represent the lengths of the three sides. Print the area to three decimal places.

$$Area = \sqrt{(s(s-a)(s-b)(s-c))}$$

```
import java.util.Scanner;        //Needed this class for input
import java.text.DecimalFormat; //Needed to format numbers to pattern

public class HeronsFormula
{
	//-------------------------------------------------------
	//Write an application that computes the area of a triangle
	//using the following Heron's formula:
	//
	//
	//          Area = srqt(s((s - a)(s - b)(s - c)))
	//
	//-------------------------------------------------------

	public static void main(String[] args)
	{
		double a=0.0,b=0.0,c=0.0,s=0.0,area=0.0;
		//variables for calculating distance of 2 points

		DecimalFormat fmt1 = new DecimalFormat("0.###");

		Scanner scan = new Scanner(System.in);//Keyboard input

		System.out.print("Enter side a: ");
		a = scan.nextDouble();
		System.out.print("Enter side b: ");
		b = scan.nextDouble();
		System.out.print("Enter side c: ");
		c = scan.nextDouble();

		s = ((a + b + c) / 2.0); //First to find s
		area = Math.sqrt(s*((s-a)*(s-b)*(s-c))); //Apply the Heron's
		//formula

		System.out.println("The area is: " + fmt1.format(area) + "
		units");//Prints out in 3 decimal places
	}
}
```

10 Design and implement an application that reads an integer value representing a year from the user. The purpose of the program is to determine if the year is a leap year (and therefore has 29 days in February) in the Gregorian calendar. A year is a leap year if it is divisible by 4, unless it is also divisible by 100 but not 400. For example, the year 2003 is not a leap year, but 2004 is. The year 1900 is not a leap year because it is divisible by 100, but the year 2000 is a leap year because even though it is divisible by 100, it is also divisible by 400. Produce an error message for any input value less than 1582 (the year the Gregorian calendar was adopted).

```
import java.util.Scanner;        //Needed this class for input

public class Gregorian
{
	//--------------------------------------------------------
	//Design and implement an application that reads an integer
	//value representing a year from the user. The purpose of
	//the program is to determine if the year is a leap year
	//(and therefore has 29 days in February) in the Gregorian
	//calendar. A year is a leap year if it is divisible by 4,
	//unless it is also divisible by 100 but not 400. For example,
	//the year 2003 is not a leap year, but 2004 is. The year 1900
	//is not a leap year because it is divisible by 100, but the
	//year 2000 is a leap year because even though it is divisible
	//by 100, it is also divisible by 400. Produce an error message
	//for any input value less than 1582 (the year the Gregorian
	//calendar was adopted).
	//
	//--------------------------------------------------------

	public static void main(String[] args)
	{
		final int GREGORIAN_YEAR = 1582; // Constant to compare to
		int x1=0;
		boolean isLeapyear = false;



		Scanner scan = new Scanner(System.in);//Keyboard input

		System.out.print("Enter the year in question (must be greater
		than or equal to 1582): ");
		x1 = scan.nextInt();
		while(x1 < GREGORIAN_YEAR)
		{
			System.out.println("Error, the year must be greater than
			or equal to 1582");
```

```
                System.out.print("Enter the year in question (must be
                greater than or equal to 1582): ");
                x1 = scan.nextInt();
        }

        if ((x1 % 4) == 0) //First test, fail this then game is over
        {
                //Check for other conditions
                //Must be either passing both 100 and 400 check
                //or failing both 100 and 400 check in order to be
                //a leap year
                if((((x1 % 100) == 0) && ((x1 % 400) == 0)) || (((x1 %
                100) != 0) && ((x1 % 400) != 0)))
                {
                        isLeapyear = true;
                }
                else
                {
                        isLeapyear = false;
                }
        }
        else //Clearly not a leap year
        {
                isLeapyear = false;
        }

        if(isLeapyear) //Display message based on isLeapyear logic
                System.out.println("The year " + x1 + " is a leap year.");
        else
                System.out.println("The year " + x1 + " is not a leap
                year.");

    }
}
```

11 Modify the solution to programming project 4.1 so that the user can evaluate multiple years. Allow the user to terminate the program using an appropriate sentinel value. Validate each input value to ensure it is greater than or equal to 1582.

```
import java.util.Scanner;       //Needed this class for input

public class Greggy2
{
    //-----------------------------------------------------
    //Design and implement an application that reads an integer
    //value representing a year from the user. The purpose of
    //the program is to determine if the year is a leap year
    //(and therefore has 29 days in February) in the Gregorian
```

```
//calendar. A year is a leap year if it is divisible by 4,
//unless it is also divisible by 100 but not 400. For example,
//the year 2003 is not a leap year, but 2004 is. The year 1900
//is not a leap year because it is divisible by 100, but the
//year 2000 is a leap year because even though it is divisible
//by 100, it is also divisible by 400. Produce an error message
//for any input value less than 1582 (the year the Gregorian
//calendar was adopted). Allows multiple entries.
//
//-----------------------------------------------------

public static void main(String[] args)
{
	final int GREGORIAN_YEAR = 1582; // Constant to compare to
	int x1=0,x2=0;
	boolean isLeapyear = false;
	//variables for calculating distance of 2 points


	Scanner scan = new Scanner(System.in);//Keyboard input

	System.out.println("Enter the year in question (must be greater
	than or equal to 1582)");
	System.out.print("Enter 9999 to quit: ");
	while((x1 = scan.nextInt()) != 9999)//Main loop for multiple
	//entries
	{
		while(x1 < GREGORIAN_YEAR)
		{
			System.out.println("Error, the year must be greater
			than  or equal to 1582");
			System.out.println("Enter the year in question (must
			be greater than or equal to 1582)");
			System.out.print("Enter 9999 to quit: ");
			x1 = scan.nextInt();
			if(x1 == 9999)
			{
				System.out.println("Good bye!!!");
				System.exit(0); //Leave the program
			}
		}

		if((x1 % 4) == 0) //First test, fail this then game is
		//over
		{
			//Check for other conditions
			//Must be either passing both 100 and 400 check
			//or failing both 100 and 400 check in order to be
			//a leap year
			if((((x1 % 100) == 0) && ((x1 % 400) == 0)) || (((x1
			% 100) != 0) && ((x1 % 400) != 0)))
			{
				isLeapyear = true;
			}
			else
```

```
                        {
                                isLeapyear = false;
                        }
                }
                else //Clearly not a leap year
                {
                        isLeapyear = false;
                }

                if(isLeapyear) //Display meassage based on isLeapyear
                //logic
                        System.out.println("The year " + x1 + " is a leap
                        year.");
                else
                        System.out.println("The year " + x1 + " is not a
                        leap year.");

                        System.out.println("Enter the year in question (must
                        be greater than or equal to 1582)");
                        System.out.print("Enter 9999 to quit: ");
        }
        System.out.println("Good bye!!!"); //Exit main while loop
    }
}
```

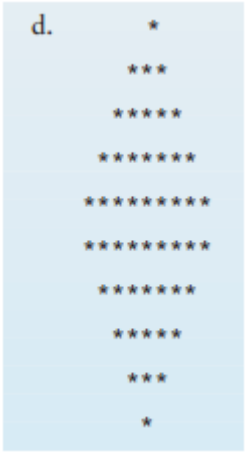
12 Design and implement an application that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Use a sentinel value to determine whether the user wants to quit. Count the number of guesses and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

```
//*******************************************************
//Author: Hieu Pham
//ID: 0953-827
//Section: 28317
//Date: 01/23/2015
//
//Generate a random integer in the range of 1 to 100
//Prompt user for an integer input
//Compare user input against random number generated
//Keep count of user's attempts
//Interact with user for continuing or quitting option.
//*******************************************************
```

```java
import java.util.Scanner; //Needed this for keyboard input

public class NumberGuessing //The class under development
{
	public static void main(String[] args) //Entry point
	{
		int x2,x1; //one for program, one for user
		int repeat=0,count=0;//Sentinel and counter
		boolean playon = true;//Flag for continue of quit status

		Scanner scan = new Scanner(System.in);//Instantiate an
						//object of Scanner class

		x2 = (int) (Math.random() * 99 + 1); //Use random method
						//of the Math class

		do
		{
		 count = 0;//Initialize for each run loop

		 do
		 {
			System.out.print("Enter your guess (a number between 1 and
			100): ");
			 x1 = scan.nextInt();//Get user input
			if (x1 == x2)//User = random
			{
				System.out.println("Right on. Congratulations!");
			}
			else if (x1 < x2)//User < random
			{
				System.out.println("Too low...");
			}
			else if (x1 > x2)//User > random
			{
				System.out.println("Too high...");
			}

			count++;//increment the trial counter
		 }while (x1 != x2);//repeat while user != random

		 //if user = random, exit loop and say something
		 System.out.printf("You took %d tries\n",count);

		 //Play another round prompting.
		 System.out.println("Do you want to play again?");

		 System.out.print("Enter 1 for yes or 9999 for no ");
		 repeat = scan.nextInt();//Get user input
		 switch(repeat)//Respond to user's choice accordingly
		 {
			case 1://Yes, user wants to play on
				playon = true;//Keep play loop active
				x2 = (int) (Math.random() * 99 + 1); //Use random
```

```
                        //method of the Math class
                        break;//Done processing
                    case 9999://No, user wants to quit
                        playon = false;//Deactivate play loop
                        break;//Done processing
                    default://Any other choice, quit
                        playon = false;//Deactivate the play loop
                        break;//Done processing
                }

            }while (playon == true);//Keep playing while playon is true
        }
}
```

13 Create modified versions of the Stars program to print the following pattern.



```java
//*********************************************************
//Author: Hieu Pham
//ID: 0953-827
//Section: 28317
//Date: 01/23/2015
//
//Modify the star program to create a diamond shape
//using asterisk characters. Implementation with swtich/case
// and for loops.
//
//
//*********************************************************

public class Stars
{
	//-------------------------------------------------
	//Prints a diamond shape using asterisk characters
	//-------------------------------------------------
	public static void main(String[] args)
	{
		final int MAX_ROWS = 10;

		System.out.println();//Give an initial blank line.
		for(int row = 1; row <= MAX_ROWS; row++)//Top half of diamond
		{
			switch(row)
			{
				case 1:
					System.out.println("    *    ");
					break;
				case 3:
					System.out.println("   ***   ");
					break;
				case 5:
					System.out.println("  *****  ");
					break;
				case 7:
					System.out.println(" ******* ");
					break;
				case 9:
					System.out.println("*********");
					break;
			}
		}

		for(int row = (MAX_ROWS-1); row >= 1; row--)//Bottom half
		{
			switch(row)
			{
				case 1:
					System.out.println("    *    ");
					break;
				case 3:
					System.out.println("   ***   ");
					break;
```

```
                        case 5:
                                System.out.println("  *****  ");
                                break;
                        case 7:
                                System.out.println(" ******* ");
                                break;
                        case 9:
                                System.out.println("*********");
                                break;
                    }
                }

        }
}
```