

Control Flow Project

The following instructions and exercises constitute the project for understanding and applying Java Control Flow statements. There will be some instructions that involve classes and/or operation that have not been covered yet. In those cases code will be provided for basic operations, but you are encouraged to use online resources to obtain further documentation and examples.

Overview

You will be building an interactive command-line program which will take input from the user, process it, and then display the results. It will be built in phases and will begin from a presentation-only system to a fully function request-response system.

First, create a simple Java project that prints "Hello World". This will verify your environment, settings, and ability to run a project on the command line. Depending on build tools and/or IDE this process can be done from the actual command line or in a console output window. The instructions assume the use of an IDE such as Eclipse or Spring Tool Suite; if a plain editor and command line tools are used those steps will have to be done by the user.

Project Steps

The first step in the project is to create a separate class that will print valid characters to the user. This will be a simple utility class that is called from the `main()` method to print valid character sets to the user. The skeleton of the class should resemble the following:

```
public class AsciiChars
{
    public static void printNumbers()
    {
        // TODO: print valid numeric input
    }

    public static void printLowerCase()
    {
        // TODO: print valid lowercase alphabetic input
    }

    public static void printUpperCase()
    {
        // TODO: print valid uppercase alphabetic input
    }
}
```

Using an ASCII chart, find the range of values for each of the different characters sets above - numbers, lower case letters, and upper case letters. Then using a `for` loop, print the valid values in each set. Note that since these are `static` methods they can be called from within the `main()` method as follows:

```
// print the valid characters for input
AsciiChars.printNumbers();
AsciiChars.printUpperCaseLetters();
AsciiChars.printLowerCaseLetters();
```

Next, set up a simple interaction with the user. After printing the valid characters, ask the question **Please enter your name:** . When the user inputs their name, simply respond to them by echoing back **Hello Tom** (if they input Tom of course). See more detailed instructions below on reading input from the keyboard.

Scanner class

To read input from the console, one of the easiest ways is to use the framework class **Scanner**. More details and explanations of several different methods exist on the official documentation site, but a simple example to read a string input is:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a string: ");    // type a word and hit enter
String someString = scanner.next();
```

Implement the following:

After the name prompt, ask the user if they wish to continue to the interactive portion. If they enter 'yes' or 'y', continue. Otherwise quit the program with a message to please return later to complete the survey.

Use a loop to repeatedly ask the user some questions. What will be done is to gather some random information from the user and then generate their potential winning Powerball numbers. At the end, ask them if they would like to generate another set of numbers. Note that it may be helpful to put example responses after each question.

Do you have a red car? (yes, no):

Ask the user for the name of their favorite pet.

Ask the user for the age of their favorite pet. (NOTE: the **next()** function will not work here, look up what the appropriate **Scanner** method is to read numbers.)

What is their lucky number?

Do they have a favorite quarterback? If so what is their jersey number?

What is two-digit model year of their car?

What is the first name of their favorite actor or actress?

Enter a random number between 1 and 50.

Generating output Now that you have 6 or 7 pieces of information it's time to generate the lottery numbers. The output will 5 random numbers between 1 and 65, and the "magic ball" between 1 and 75. To accomplish this you should have 3 random integers, generated in the same way as in the previous exercises. You may also want to declare constants for the max values of each set of numbers.

Generate the "magic ball" first. It should be the user's favorite quarterback jersey number *or* their lucky number, multiplied by one of the random numbers. Make sure that the answer is less the max "magic" number of 75; if it is greater than 75 then subtract 75 from the value and that value is now the "magic ball" number.

For the 5 non-magic numbers use your choice of any of the following methods to generate values: (Note that you may have to adjust each value to correct range by adding/subtracting.)

- Find the 3rd letter of their favorite pet. Convert that character value to an integer value.
- Use the two digit model year of their car and add their lucky number to it.
- Use the random number between 1 and 50, subtracting one of the generated random numbers.
- Convert the first letter of their favorite actor/actress to an integer and use that value.
- Convert the last letter of their favorite actor/actress to an integer and use that value.
- Use the value 42.
- Use the age of their favorite pet + their car model year.
- Favorite quarterback number + age of pet + lucky number.

Once all the numbers have been calculated, print them out in this form:

```
Lottery numbers: 4, 17, 15, 52, 26 Magic ball: 22
```

Play again? After the numbers have been printed, ask the user if they would like to answer questions to generate another set of numbers. If so, repeat the entire process. If not, print a thank you message and exit.

Once the basics of this are working (happy path), go back and add error-checking for each question. If the user does not enter a valid entry, ask the question again until a valid entry is produced.

Bonus

- Remove duplicate numbers
- sort the first 5 numbers.