Programming

Changing a Program

In this lesson the program, “Greetings”, is changed to prompt the user for input, and then greet the user. This necessitates explaining importing two classes from the Java library and handling exceptions.

Lesson Outcomes

By the end of the lesson you should be able to:

- Read input from the standard input stream

Lesson Notes

Remember that System.out is an object object that represents the standard output. There is another object in the System class which, like the out object, is created by the Java Virtual Machine when your programme starts running. It’s called an “in” andrepresents the standard input stream, which is usually the keyboard. So, any input that is typed in by the user is collected by the :”System.in” object.

The “System.in” object is not as easy to use as the “System.out” object. It has a method with the name “read” which only give us one byte each time we ask the, System.in” object to perform its read method. This gives us two challenges.

Firstly, the read method gives the ASCII (8 bit) code for each character but Java uses Unicode (16 bit) for its characters.

Secondly, the read method returns one character at a time. It’s better to be able to read all the characters together as a string, not separately.

Fortunately, there are other classes in the Java library which can help us solves these problems. The Java library is organized into what we call “packages” – collections of related classes. In this lesson we use the classes “InputStreamReader” and “BufferedReader” which are in the java.io package. To be able to use these classes we must import them using the following statement:

```
import java.io.*;
```

The class “InputStreamReader”.

Objects of this class convert a stream of bytes into a stream of characters. We can create an InputStreamReader object by writing “new InputStreamReader” but we must remember to save the new object’s address in memory to the “reader” object variable. But, this object still doesn’t do what we want, because it only gives us one character at a time, and we would like a whole string of characters.

The class “BufferedReader”

Objects of this class can collect the individual characters from an InputStreamReader object and give us one whole line of input at a time. Remember, we must declare an object variable to store its address. Its type is BufferedReader and we assign its address (=) to keyboard .

We use the “readLine” method of the BufferedReader object to input one line of text at a time from the standard input stream (keyboard). This method returns a line of input to us as a String object. In Java all strings are objects. Not all programming languages see strings in this way. With other languages, like Delphi, strings are more like primitive types. This means we can’t do arithmetic with strings, but we can use methods.

Something could go wrong when you read input. Java calls something that goes wrong, or something out of the ordinary, an “exception”. The compiler insists that you take notice of exceptions and do something about them. You can either “catch” the exception, or “throw” an exception. To throw any exception that occured when reading input, add “throws IOException” to the end of the method header. This means we’re saying “Yes, I know something can go wrong, but I don’t want to try to fix it. Pass the exception on to the Java Virtual Machine”

Task

1. Write a program, called “Story”, that will prompt the user to enter a colour, a vegetable and an animal. Use these three words in a short story. For example, if the user enters “green”, “potato” and “pig”, the story could be something like:

   After my sister ate the potato,
   she turned green
   and hugged the pig.