CP213 Lesson 2

- for formatted printing use the printf method

- Println, print and printf are methods from the PrintStream class

- printf method can have unlimited arguments, the first argument is always a string that contains format information

Ex.

package lesson02;

public class FormatedOutput {

public static void main(String args[]) {

double yourBill = 39.99;

System.out.print("$");

System.out.printf("%6.2f", yourBill);

System.out.println(" total");

}

}

>> $ 39.99 total

* “%6.2f” is the output specifier
* % indicates it’s a format specifier
* The 6 says to display up to 6 right-justified characters
* .2 means to display 2 decimals
* F means it’s a float
* To left justify add a “-“ character after the percentage
* % is the same as {} in python

Ex.

System.out.printf("$%6.2f for each %s. %n", salePrice, name);

>>$199.99 for each Android phone

is an excellent price

-%n prints a new line character

Common format specifiers

* %c character
* %d decimal (integer) number (base 10)
* %e exponential floating-point number
* %f floating-point number
* %i integer (base 10)
* %o octal number (base 8)
* %s String
* %u unsigned decimal (integer) number
* %x number in hexadecimal (base 16)
* %t formats date/time
* %% print a percent sign
* \% print a percent sign

To print money you can use

NumberFormat moneyFormatter = NumberFormat.getCurrencyInstance();

System.out.println(moneyFormatter.format(39.89999));

>>$39.90

1. 2 decimals
2. Second decimal is rounded
3. Adds a dollar sign to the front

* You can import a locale and print different currency symbols as well

Ex.

import java.text.NumberFormat;

import java.util.Locale;

moneyFormatter = NumberFormat.getCurrencyInstance(Locale.GERMANY);

system.out.print(moneyFormatter.format(39.89999);

>>39,90 €

To make a package, group all the classes together into a single directory (folder) and add the following statement at the beginning of each class file:

**Package package\_name**

**The Scanner Object**

* To create a scanner object first import the scanner using “import java.util.Scanner;”
* Then write “Scanner input = new Scanner(System.in);”
* nextInt reads one int value typed in at the keyboard and can be assigned to a var ex. Int x = input.nextInt();
* same thing for nextDouble, nextFloat etc…
* the next method takes the whole input until a whitespace
* nextLine() takes an entire line
* both next and nextLine return strings
* You can also use Scanner to read files

Ex.

Scanner inputFile= new Scanner(new FileInputStream(fileName));

FileInputStream acts like system.in but you have to instantiate it with the “new” keyword

**Wrapper classes**

* The mechanism used to convert primitive types to object and vice-versa is called wrapping
* You can make primitive data types into object for

Ex.

Integer object = new Integer(2020); // primitive to Object

int value = object.intValue(); //Object to primitive

Ex.

ArrayList<Integer> arrayList = new ArrayList<>();

for (int i = 1; i < 10; i++) {

arrayList.add(i); // int to Integer

}

}

arrayList is a list of Integer object, not a list of primitive types and int values. The Java compiler automatically creates an Integer object from I and adds it to the arrayList.

**Keywords**

Autoboxing

* Introduced in Java 5.0 autoboxing is a feature that allows us to convert primitives to objects (the other way around is called unboxing). Converting a primitive to its object counterpart can be useful because it allows you to call upon many methods such as the .charAt() method for the Character object.

DecimalFormat class

FileInputStream class

format()

format specifier

getCurrencyInstance()

IntValue

Locale

* When using the NumberFormat class you can format your numbers in your program as currency. There are many different currencies around the world and they all have different signs for example the Canadian Dollar uses the $ but the Euro does not. By passing the Locale.COUNTRY argument to the getCurrencyInstance method of the NumberFormat class you can output using the proper currency sign. You must import the Locale class into your code.

nextInt(), nextDouble(), etc.

nextLine() and hasNextLine()

NubmerFormat class

print, println, printf

* Three ways to print output to the console, print() prints output on the current line, println() prints output as well as a newline character and printf allows you to format your output. Formatting output is helpful to output multiple lines that can all have the same spacing or to round numbers or properly format $ signs.

PrintStream

Scanner class

* Similar to input() in python the Scanner class is how the programmer can get input from the user in java. To use it first create a Scanner object, then use Scanner methods such as .next(), .nextInt(), .nextDouble() and store them in variables to get appropriate user data. You must import the Scanner class into your code.

specifier justification and “\n”

Unboxing

useDelimiter()

* Method in the Scanner class when getting string input. Using the useDelimiter(*Seperator*) will split input up based on whatever the delimiter is. For example for a Scanner object in, using in.useDelimiter(“,”) means that you can use the in.next() method and only get the string up to and not including the comma. The comma will not be taken at all. This could be very useful when interacting with a CSV file that only uses commas as delimiters.

valueOf

Wrapping