

LECTURE 1 & 2 – PROGRAMMING 1B PROG6112

Warming up!

Task 1 (20 minutes)

Create a Java application that uses printf statements. Consult the Appendix on page 931. Create a class called myPrintfExample and copy code from Figure C-3. Run the code and explain the output.

```
public class RoundingDemo1
{
    public static void main(String[] args)
    {
        double answer = 2.20 - 2.00;
        boolean isEqual;
        isEqual = answer == 0.20;
        System.out.println("Before conversion");
        System.out.println("answer is " + answer);
        System.out.println("isEqual is " + isEqual);
        answer = answer * 100;
        answer = answer + 0.5;
        answer = (int) answer;
        answer = answer / 100;
        isEqual = answer == 0.20;
        System.out.println("After conversion");
        System.out.println("answer is " + answer);
        System.out.println("isEqual is " + isEqual);
    }
}
```

Discover alt+shift+F and alt+shift+left/right arrow key to indent or unindent selected lines. Illustrate when copying the code from the pdf textbook.

Display the strings with 5 spaces to the right of the two strings:

```
String string1 = "hello";
String string2 = "there";
System.out.printf("%-10s%-10s", string1, string2);
```

Display the strings with 5 spaces to the left of the two strings:

```
String string1 = "hello";
String string2 = "there";
System.out.printf("%10s%10s", string1, string2);
```

Create the following output: 56 78.90 56 78.90

Display the value of the first argument, x, in a field with a size of 6, and then it displays the second argument, y, in a field with a size of 6 with two decimal places. Then, the value of x is displayed again, followed by the value of y

```
int x = 56;
double y = 78.9;
System.out.printf("%1$6d%2$6.2f%1$6d%2$6.2f", x, y);
```

Create the toString() method that returns a string from the class.

```
public String toString()
```

```
{
    return String.format("%d:%02d%02d %s", getters for four parameters);
}
```

Task 2 (30 minutes)

Write a Java application that will create a pyramid of asterisks. First think through the algorithm to produce the pyramid. The input parameters are “How many spaces must first star be printed away from left margin and the second parameter of the printStars method will be the number of lines to draw the pyramid :

printStars (int blanks, int starsInLine), which are 30 and 10 in this example.

```

      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
*****

```

Task 3 (10 minutes)

Create the class NumberList and copy code from C-4.

```
public class NumberList
{
    public static void main(String[] args)
    {
        double[] list = {0.20, 2.00, 2.20, 22.22,22.20, 222.00, 222.22};
        int x;
        for(x = 0; x < list.length; ++x)
            System.out.println(list[x]);
    }
}

public class ConversionCharacterExamples
{
    public static void main(String[] args)
    {
        int age = 23;
        double money = 123.45;
        System.out.printf("Age is %d\n", age);
        System.out.printf("Money is $%f\n", money);
        System.out.printf
            ("Age is %d and money is $%f\n", age, money);
    }
}
```

Task 4 (10 minutes)

Copy code of Figure C-8. Arrange the indent in an automatic manner.

Discuss the orientation of the [] angle brackets.

```
public class NumberList2
{
    public static void main(String[] args)
    {
        double[] list = {0.20, 2.00, 2.20, 22.22,
                        22.20, 222.00, 222.22};
        // or double[] list = new double[7];
        int x;
        for(x = 0; x < list.length; ++x)
            System.out.printf("%6.2f\n", list[x]);
    }
}
```

Discuss the static declaration versus the dynamic declaration using new. Example: //
or double[] list = new double[7];

Task 5 (20 minutes)

Arrays from numbers to classes

5.1 WRITE A JAVA APPLICATION TO ACCOMPLISH THE FOLLOWING:

- a) Write Java statements to declare an array that will store the monthly average rainfall values (double) for a city for one year.
- b) Write Java statements to declare another array and initialize it with the names of the months of the year.

5.2 NOW FILL THE ARRAY WITH FICTITIOUS VALUES AND:

- c) Write Java statements to determine and display the highest average rainfall value, and the month in which it occurred.

5.3 WRITE A JAVA APPLICATION FOR:

- d) Write the declaration of an integer array myTable that has 3 rows and 4 columns.
- e) Initialize the array myTable with random numbers in the range 50 to 100 inclusively.
- f) The user enters a value, rowNum, representing a row of the array myTable. Write the statements to calculate and display the sum of the elements in that row.

Task 6 (20 minutes)

Arrays from numbers to classes

6.1 Write a Java Application with a method called “daysInMonth”, to be called from the main method, which resides in the Java application class with the following functionality.

- a. The method will have two parameters month (int) and year (int).
- b. The method will return an integer value, which will be the number of days of a month for that year, e.g. `daysInMonth(6,2000)` will return a value of 30 and so on. If the year is a leap year then for the month of February should return 29 days. The formula for a (Gregorian) leap year must be used, i.e. the year must be divisible by 4, but not by a 100 (except if divisible by 400)

Task 7 (30 minutes)

Arrays from numbers to classes

7.1 Write an application to simulate the rolling of two dice. The application should use an object of class Random once to roll the first die and again to roll the second die. Each die can show an integer value from 1 to 6. Your application should roll the dice 40,000 times. The following prerequisites are of the essence:

- a. Use a two-dimensional array called `twoDim` to tally the number of times each combination of the two die are thrown in that order only.
- b. Display the results in tabular format, by calling a `Display` method with the array as parameter. These results will be the number of times a combination was thrown, e.g. if a 5 is thrown thrice using `Die_1` and a 4 is thrown thrice using `Die_2`, then the Java application must keep track of the number of times (three in this case) 5 and 4 occurred together, but which is not the same when 4 and 5 were thrown.

7.2 You are starting a business and need to write a program to keep track of all the customers.

The customer class consists of attributes, `Custnumb` (Integer) and `CustName` (String). Assume the CUSTOMER class has been declared as follows:

Customer
-CustNumb : int -CustName : String
<<constructor>>+Customer(CNumb : int, CName : String) +setCustNumb(cnumb : int) : void +setCustName(cname : String) : void +getCustNumb() : int +getCustName() : String +toString() : String

ONLY write Java code for the Java application considering the following prerequisites:

1. Use the `JOptionPane` class as input device.

2. Declare an array for Customer, called CustList, which must be globally visible by all methods within the Java application.
3. Allow the user to specify how many Customers he/she wants to capture during runtime.
4. Use a for-loop and write code to capture Customers within CustList making use of the constructor in your code.
5. Write a separate for-loop to display these customers within CustList.