Tshwane University of Technology

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

PPAF05D/TROF05D

Unit 6 SOLUTIONS to Activities

User-defined static methods in Java

DEVELOPED: 2019-2021

Activity 1: Write programs with value-returning methods

a) Complete the value-returning method for the following program. The method should round the value that is sent as an argument to the required number of decimal values. (You need to create the program. Type the provided code for the main() method, and complete the code for the RoundToDecimals() method.)

```
package roundto;
import java.util.Scanner;
public class RoundTo {

    public static double RoundToDecimals(double rTheValue, int iNum)
    {
                       Complete this code.
    } //end RoundToDecimals

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        double rFloatValue;
        int iNumDecimals;
        System.out.print("Enter floating point value: ");
        rFloatValue = kb.nextDouble();
        System.out.print("Round to how many decimal values?: ");
        iNumDecimals = kb.nextInt();
        rFloatValue = roundToDecimals(rFloatValue, iNumDecimals);
        System.out.println("The rounded value is: " + rFloatValue);
    } //end main

}// end class
```

Solution Activity 1a)

```
    public static double roundToDecimals(double rTheValue, int iNum)
    {
        double rFactor = Math.pow(10, iNum);
        rTheValue = Math.round(rTheValue * rFactor)/rFactor;
        return rTheValue;
    } //end RoundToDecimals
```

b) Complete the value-returning method for the following program. The method should calculate the factorial of the value that is sent as an argument. (You need to create the program. Type the provided code for the main() method, and complete the code for the factorial () method.)

```
package calcfactorial;
import java.util.Scanner;
public class CalcFactorial {

    public static double factorial(int iValue)
    {
                        Complete this code.
    } //end factorial

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int iNumber;
        double rFact;
        System.out.println("I will calculate the factorial.");
        System.out.print("Enter a number: ");
        iNumber = keyboard.nextInt();
        rFact = factorial(iNumber);
        System.out.println("The factorial of " + iNumber + " is :" + rFact);
    } //end main

}
```

Solution Activity 1b)

```
public static double factorial(int iValue)
{
    double rLocal = 1;
    for (int iCounter = 2; iCounter <= iValue; iCounter ++)
    {
        rLocal = rLocal * iCounter;
    }
    return rLocal;
} //end factorial
```

c) Complete the value-returning method for the following program. The method should ask the user how many random digits a number should contain. Tip: Refer to example 2 – the example created a value containing 4 digits that was created randomly – you can adapt that program so that the number of digits is a parameter of the method. (You need to create the program. Type the provided code for the main() method, and complete the code for the getRandNum() method.)

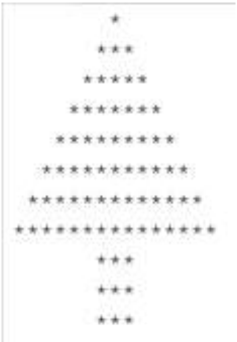
```
package randomnumberdigits;
import java.util.Random;
import java.util.Scanner;
public class RandomNumberDigits {
    {

                         Complete this code.

    } //end getRandNum

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        int iNumRandom, iTheRandomNumber;
        System.out.print("How many random digits do you want?: ");
        iNumRandom = kb.nextInt();
        iTheRandomNumber = getRandNum(iNumRandom);
        System.out.println("The number is: " + iTheRandomNumber);
    } //end main
        public static int getRandNum(int iHowManyDigits)
}//end class
```

Solution Activity 1c)

```
    public static int getRandNum(int iHowManyDigits)
    {
        Random randomize = new Random();
        double rFactor = Math.pow(10, iHowManyDigits - 1);
        int iFirstDigit = randomize.nextInt(9) + 1 ; //a digit between 1 and 9
        int iOtherDigits = (int) (Math.random() * rFactor);
        return (int) (iFirstDigit * rFactor) + iOtherDigits;
    } //end getRandNum
```

Activity 2: Write programs with void methods

a) Create a Java program that will draw a tree as displayed in the screenshot alongside. A framework of the program is provided. You need to complete the code as indicated. (The tree has 8 rows of stars. The longest row has 15 stars.)



```
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
      ***
      ***
      ***
```

```
package drawtree;
public class DrawTree {

    public static void drawBlanks(int iNum)
    {
```

Complete the code to draw the required number of spaces.

```
    } //drawBlanks

    public static void drawStars(int iNum)
    {
```

Complete the code to draw the required number of stars.

```
    } //drawStars
    public static void goNextLine()
    {
        System.out.println();
    } //goNextLine

    public static void main(String[] args) {
        int iCountLines, iCountBlanks = 7, iCountStars = 1;
        //draw the branches
        for (iCountLines = 1; iCountLines <= 8; iCountLines++)
        {
            drawBlanks(iCountBlanks);
            drawStars(iCountStars);
            goNextLine();
            // decrease iCountBlanks by 1
            // increase iCountStars by 2
        }
         //draw the stem
```

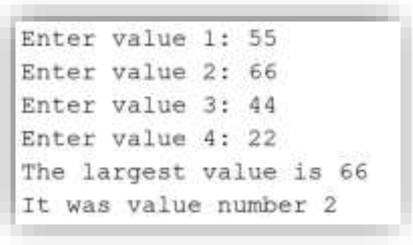
Complete the code to draw the stem of the tree.

```
    } //end main

} //end class
```

Solution Activity 2a)

```
package drawtree;
public class DrawTree {

    public static void drawBlanks(int iNum)
    {
        for (int iCounter =1; iCounter <= iNum; iCounter ++)
            System.out.print(" ");
    } //drawBlanks

    public static void drawStars(int iNum)
    {
        for (int iCounter =1; iCounter <= iNum; iCounter ++)
            System.out.print("*");
    } //drawStars

    public static void goNextLine()
    {
        System.out.println();
    } //goNextLine

    public static void main(String[] args) {
        int iCountLines, iCountBlanks = 7, iCountStars = 1;
        //draw the branches
        for (iCountLines = 1; iCountLines <= 8; iCountLines++)
        {
            drawBlanks(iCountBlanks);
            drawStars(iCountStars);
            goNextLine();
            iCountBlanks--;
            iCountStars += 2;
        }
         //draw the stem
        for (iCountLines = 1; iCountLines <= 3; iCountLines++)
        {
            drawBlanks(6);
            drawStars(3);
            goNextLine();
        }
    } //end main

}
```

b) Complete the following program. The purpose of the program is to read 4 integer values, and then display the value as well as the position (index) of the largest number.



```
Enter value 1: 55
Enter value 2: 66
Enter value 3: 44
Enter value 4: 22
The largest value is 66
It was value number 2
```

```
package biggestof4values;
import java.util.Scanner;
public class BiggestOf4Values {

    public static void displayLargest(int i1, int i2, int i3, int i4)
    {
```

Complete the code to determine the largest of the 4 values as well as the position of the value in the list.

```
        System.out.println("The largest value is " + iLargest);
        System.out.println("It was value number " + iLargestIndex);
    } // end displayLargest

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        int iVal1, iVal2, iVal3, iVal4;

        System.out.print("Enter value 1: ");
        iVal1 = kb.nextInt();
        System.out.print("Enter value 2: ");
        iVal2 = kb.nextInt();
        System.out.print("Enter value 3: ");
        iVal3 = kb.nextInt();
        System.out.print("Enter value 4: ");
        iVal4 = kb.nextInt();
        displayLargest(iVal1, iVal2, iVal3, iVal4);
    } //end main
}
```

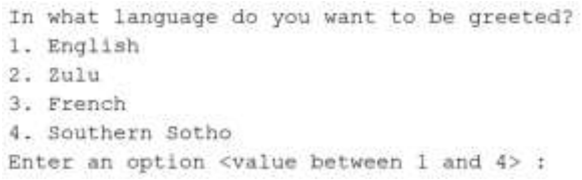
Solution Acivity 2b)

```
package biggestof4values;
import java.util.Scanner;
public class BiggestOf4Values {

    public static void displayLargest(int i1, int i2, int i3, int i4)
    {
        int iLargestIndex = 1;
        int iLargest = i1;
        if (i2 > iLargest){
            iLargestIndex = 2;
            iLargest = i2;
        }
        if (i3 > iLargest){
            iLargestIndex = 3;
            iLargest = i3;
        }
        if (i4 > iLargest){
            iLargestIndex = 4;
            iLargest = i4;
        }
        System.out.println("The largest value is " + iLargest);
        System.out.println("It was value number " + iLargestIndex);
    } // end displayLargest

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        int iVal1, iVal2, iVal3, iVal4;

        System.out.print("Enter value 1: ");
        iVal1 = kb.nextInt();
        System.out.print("Enter value 2: ");
        iVal2 = kb.nextInt();
        System.out.print("Enter value 3: ");
        iVal3 = kb.nextInt();
        System.out.print("Enter value 4: ");
        iVal4 = kb.nextInt();
        displayLargest(iVal1, iVal2, iVal3, iVal4);
    } //end main
}
```

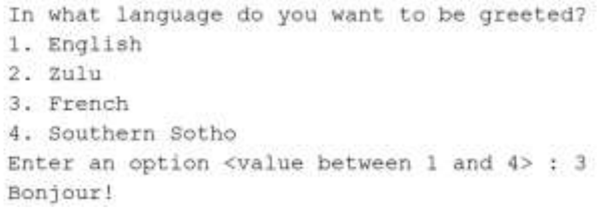
Activity 3: Write programs with value-returning and void methods

a) Create a Java program that will do the following.

- Use a value-returning method getMenuOption to display the menue, and read a menu option until the option entered is a valid value. See Void method example 2: Display the name of a season for an example of such a method.
- Use a void method that receives the menu option (1 – 4), and displays a greeting in the chosen language.



```
In what language do you want to be greeted?
1. English
2. Zulu
3. French
4. Southern Sotho
Enter an option <value between 1 and 4> :
```



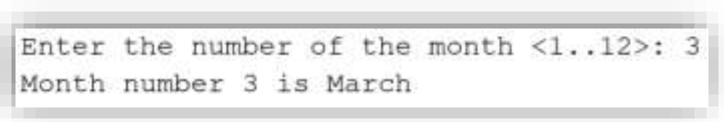
```
In what language do you want to be greeted?
1. English
2. Zulu
3. French
4. Southern Sotho
Enter an option <value between 1 and 4> : 3
Bonjour!
```

Solution Activity 3a)

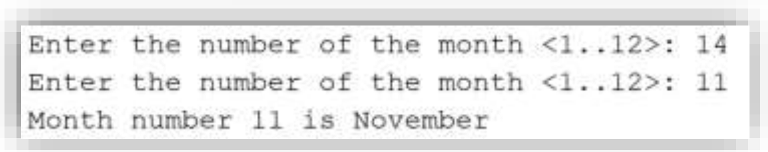
```
package languageoption;
import java.util.Scanner;
public class LanguageOption {

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        int iMenuOption = getMenuOption();
        GreetMe(iMenuOption);
    } //end main

        public static int getMenuOption()
    {
       Scanner kb = new Scanner(System.in);
       int iOption;
       System.out.println("In what language do you want to be greeted?\n1. English"+
                          "\n2. Zulu \n3. French \n4. Southern Sotho");
       do
       {
           System.out.print("Enter an option <value between 1 and 4> : ");
           iOption = kb.nextInt();
       }
       while (iOption < 1 || iOption > 4);
       return iOption;
    } //end getValidMonth

    public static void GreetMe(int iChoice)
    {
        switch (iChoice)
        {
            case 1: System.out.println("Good day!"); break;
            case 2: System.out.println("Sawubona!"); break;
            case 3: System.out.println("Bonjour!"); break;
            case 4: System.out.println("Khotsong!");
        }
    } //end GreetMe
}
```

b) Create a Java prorgam that will read a month number (1 – 12), and then display the name of the corresponding month.

- Use a value-returning method to read a number until it is valid.
- Use a void method to display the correct name of the month.



```
Enter the number of the month <1..12>: 3
Month number 3 is March
```



```
Enter the number of the month <1..12>: 14
Enter the number of the month <1..12>: 11
Month number 11 is November
```

Solution Activiy 3b)

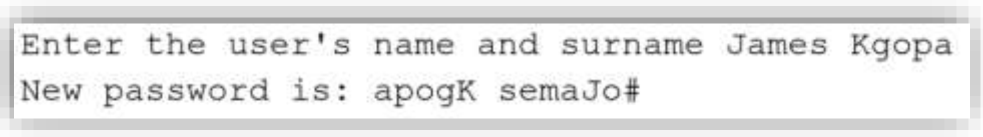
```
package convertmonthnumtomonthname;
import java.util.Scanner;
public class ConvertMonthNumTOMonthName {

    public static int readMonthNumber()
    {
        int iMonNum;
        Scanner kb = new Scanner(System.in);
        do {
            System.out.print("Enter the number of the month <1..12>: ");
            iMonNum = kb.nextInt();
        } while (iMonNum <= 0 || iMonNum > 12);
        return iMonNum;
    } // end readMonthNumber

    public static void displayMonthName(int iNumber)
    {
       String sMonthName = "";
       switch(iNumber)
       { //begin switch
           case 1: sMonthName = "January"; break;
           case 2: sMonthName = "February"; break;
           case 3: sMonthName = "March"; break;
           case 4: sMonthName = "April"; break;
           case 5: sMonthName = "May"; break;
           case 6: sMonthName = "June"; break;
           case 7: sMonthName = "July"; break;
           case 8: sMonthName = "August"; break;
           case 9: sMonthName = "September"; break;
           case 10: sMonthName = "October"; break;
           case 11: sMonthName = "November"; break;
           case 12: sMonthName = "December"; break;
       } //end switch
       System.out.println("Month number "+ iNumber + " is " + sMonthName);
    } //end displayMonthName

    public static void main(String[] args) {
        int iMonthNum;
        iMonthNum = readMonthNumber();
        displayMonthName(iMonthNum);
    } //end main
}
```

c) Write a Java program to create a password for a user according to the following rules:

- Read the name and surname. Use a value-returning method to reverse the order of the letters in the string. (For example the name and surname "James Kgopa" should become " apogK semaJ". (See Unit 5 Activity 22.)
- The password should be the reversed name and surname, plus the third character of the reversed string, plus a random special character between ASCII values 33 and 47 – both included. Use a value-returning method to generate this special character.

This is the code for the main() method.

```
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        String sSentence;
        System.out.print("Enter the user's name and surname ");
        sSentence = keyboard.nextLine();
        createPassword(sSentence);
    } //end main
```



```
Enter the user's name and surname James Kgopa
New password is: apogK semaJo#
```

Solution Actiity 3c)

```
package userpassword;
import java.util.Scanner;
import java.util.Random;
public class UserPassword {

    public static String reverseStr(String sSentence)
    {
        String sLocal = "";
        int iCounter;
        for (iCounter = sSentence.length()-1; iCounter >= 0; iCounter --)
        {
            sLocal = sLocal + sSentence.charAt(iCounter);
        }
        return sLocal;
    } //end reverseStr

    public static char returnRandCharBetween (int iLowASCII, int iUppASCII)
    {
        Random randomizer = new Random();
        int iRange = (iUppASCII - iLowASCII);
        return (char) (randomizer.nextInt(iRange + 1) + iLowASCII);
    }//returnRandCharBetween
```

```java
    public static void createPassword(String sSentence)
    {
        String sPassw;
        sPassw = reverseStr(sSentence);
        sPassw = sPassw + sPassw.charAt(2) + returnRandCharBetween(33, 47) ;
        System.out.println("New password is: " + sPassw);
    } //end createPassword

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        String sSentence;
        System.out.print("Enter the user's name and surname ");
        sSentence = keyboard.nextLine();
        createPassword(sSentence);
    } //end main
}
```