SOFTWARE DESIGN
& PROGRAM DEVELOPMENT

Java Lab

Week 5 – Output using printf

IMPORTANT! Save all your work to a safe location such as oneDrive.

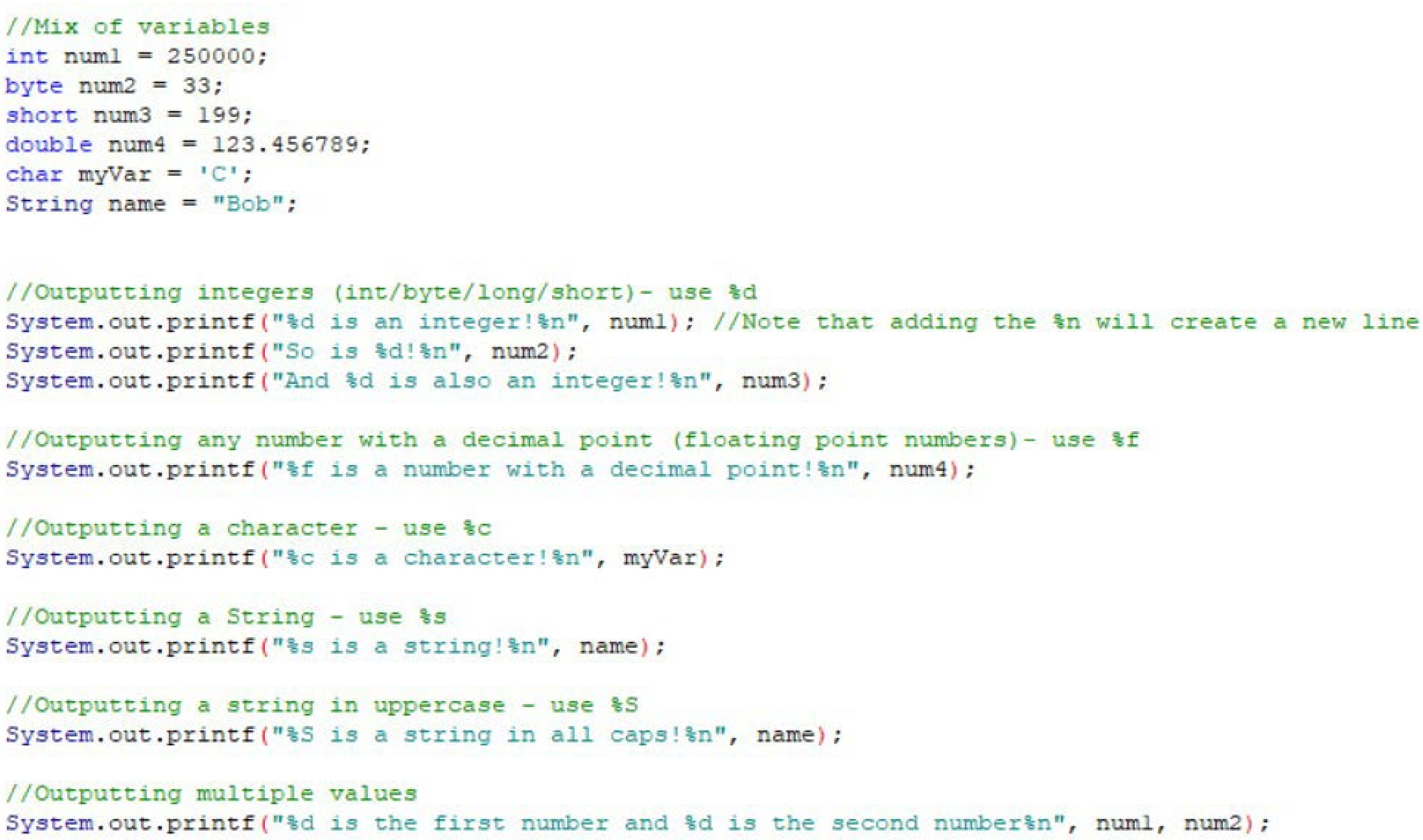
Create a folder for SDPD into which you will save all your work for this module, arranged how you wish. Ideally you should create a folder each week for your lab exercises. Note that you should create **a separate file** for each exercise.

Printf

Quick Reference

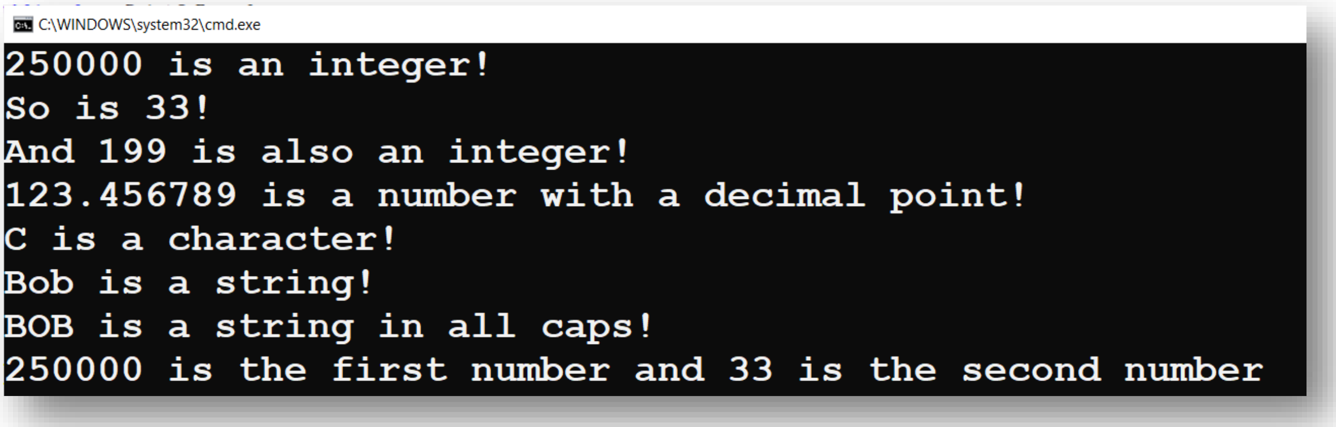
printf	
`%d`	Integer (decimal base 10)
`%c`	Character
`%f`	Floating point values (Floats and Doubles)
`%s`	Strings
`%S`	Strings(Using the uppercase S will convert string to all caps)
Formatting Numbers with Decimal Points or commas	
`%.1f`	Specify number of decimal places in output (for example, 2.357123 will become 2.4)
`%,f`	Use commas to separate thousands in output (for example: 56345 will become 56,345)
`%,.2f`	Use commas and specify decimal points for output (for example: 1234.5678 will become 1,234.57

Examples of output using Printf (See the file Printf_Examples in the “Code Samples” section in moodle)



```
//Mix of variables
int num1 = 250000;
byte num2 = 33;
short num3 = 199;
double num4 = 123.456789;
char myVar = 'C';
String name = "Bob";


//Outputting integers (int/byte/long/short)- use %d
System.out.printf("%d is an integer!%n", num1); //Note that adding the %n will create a new line
System.out.printf("So is %d!%n", num2);
System.out.printf("And %d is also an integer!%n", num3);

//Outputting any number with a decimal point (floating point numbers)- use %f
System.out.printf("%f is a number with a decimal point!%n", num4);

//Outputting a character - use %c
System.out.printf("%c is a character!%n", myVar);

//Outputting a String - use %s
System.out.printf("%s is a string!%n", name);

//Outputting a string in uppercase - use %S
System.out.printf("%S is a string in all caps!%n", name);

//Outputting multiple values
System.out.printf("%d is the first number and %d is the second number%n", num1, num2);
```

Produces the following output:



```
250000 is an integer!
So is 33!
And 199 is also an integer!
123.456789 is a number with a decimal point!
C is a character!
Bob is a string!
BOB is a string in all caps!
250000 is the first number and 33 is the second number
```

Exercise 1

Goal: Create a program in Java, using printf to format the output using integer variables.

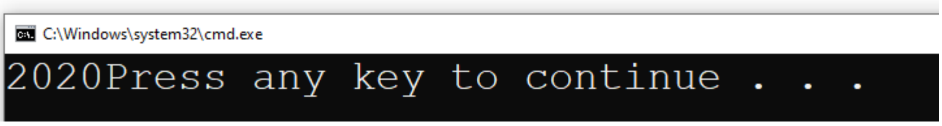
Create a new a Java program called JavaPrintf1. Create an int variable called year with the value as shown:

```
int year = 2020;
```

Output the value using printf (in place of println) as shown:

```
System.out.printf("%d", year);
```

Your output should be similar to as shown below:



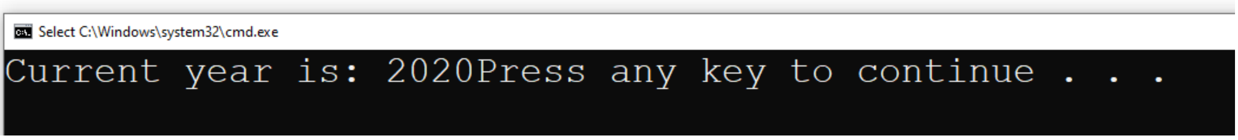
```
2020Press any key to continue . . .
```

Note that the format specifier %d refers to base 10 decimal numbers. The printf() method accepts all the integers available in the language; byte, short, int and long using `%d`

Amend your output so that it is as follows:

```
System.out.printf("Current year is: %d", year);
```

Output should be similar to as shown:



```
Current year is: 2020Press any key to continue . . .
```

Add a second int variable to your code called month, eg:

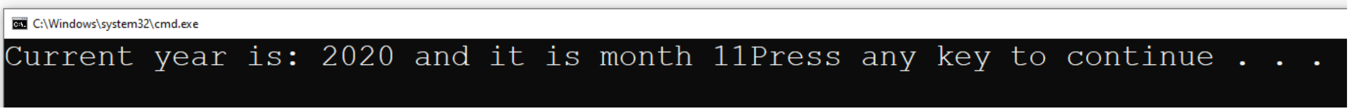
```
int year = 2020;
int month = 11;
```

Amend your printf output line:

```
System.out.printf("Current year is: %d and it is month %d", year, month);
```

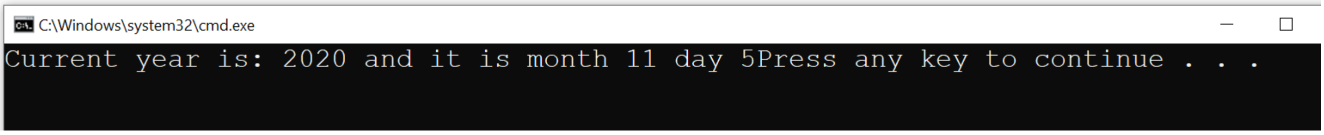
Note that the first %d (known as the *format specifier*) refers to the year variable and the second format specifier %d refers to the month variable.

The console output should be similar to as shown:



```
Current year is: 2020 and it is month 11Press any key to continue . . .
```

Add a third int variable called day that has a value of 5. Amend your code so that the following output is produced using printf:



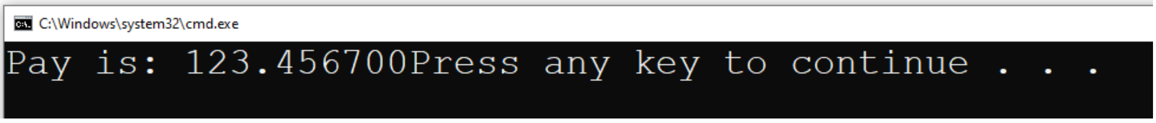
```
Current year is: 2020 and it is month 11 day 5Press any key to continue . . .
```

Exercise 2

Goal: Create a program in Java, using printf to format the output, using double variables.

Create a new a Java program called JavaPrintf2. Create a double variable called pay with the value 123.4567. Output the value using printf to produce the output shown.

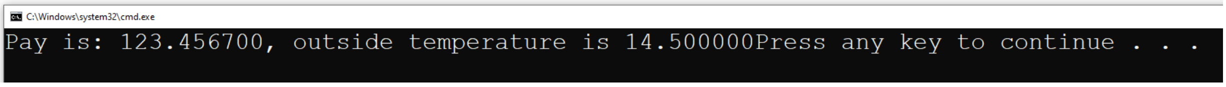
To format a float number with printf, use ***%f***



```
Pay is: 123.456700Press any key to continue . . .
```

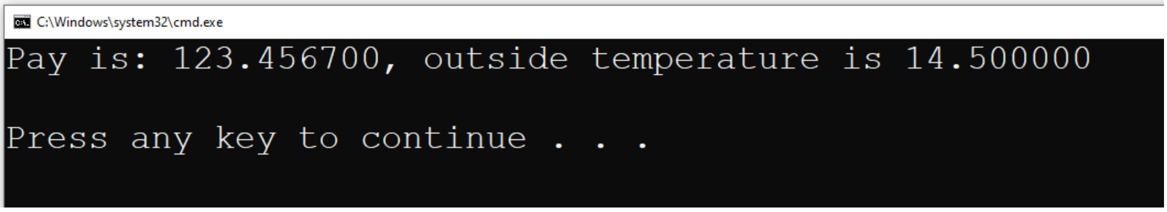
Add an additional double variable called temperature and amend your output so that it is as follows:

Output should be similar to as shown:



```
Pay is: 123.456700, outside temperature is 14.500000Press any key to continue . . .
```

Add **\n** as required to add new lines:



```
Pay is: 123.456700, outside temperature is 14.500000

Press any key to continue . . .
```

Exercise 3

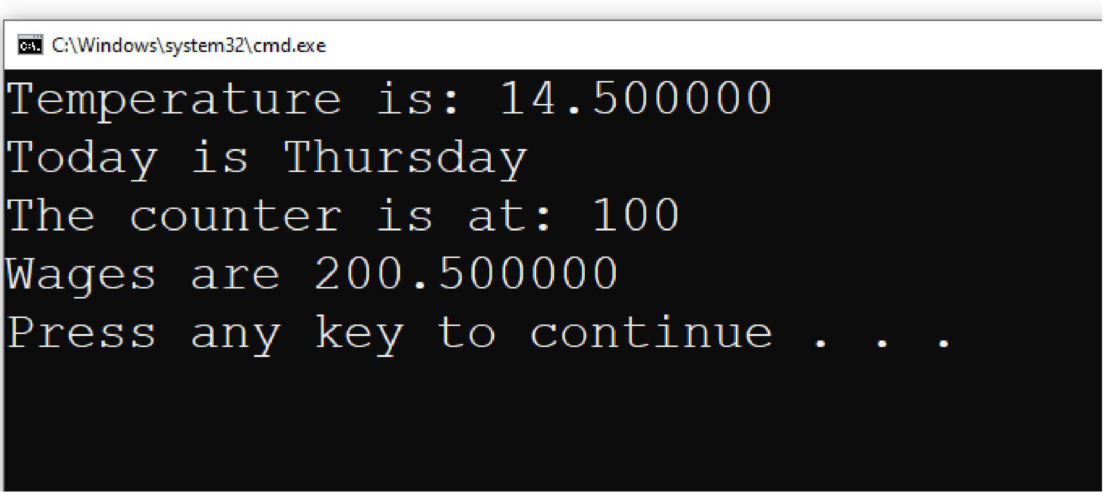
Goal: Create a program in Java, using printf to format the output, using double, int and String variables.

Create a new a Java program called JavaPrintf3. Create the following variables:

A double to store temperature
A String to store the day
An integer to store a counter number
A float to store a value for wages

(See output below for values)

Using a single printf statement, your program should produce an output similar to as shown below:



```
Temperature is: 14.500000
Today is Thursday
The counter is at: 100
Wages are 200.500000
Press any key to continue . . .
```

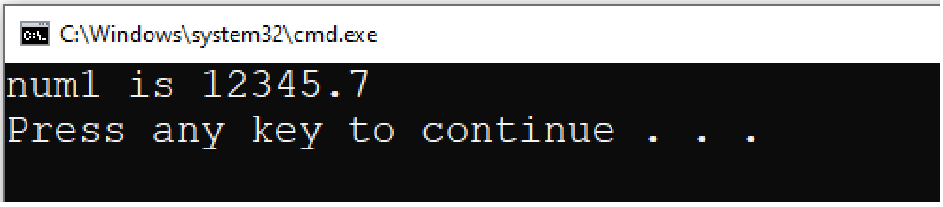
Exercise 4

Goal: Create a program in Java, using printf to format the output as required.

Create a new a Java program called JavaPrintf4.

- Assume the double variable num1 contains the value 12345.6789.

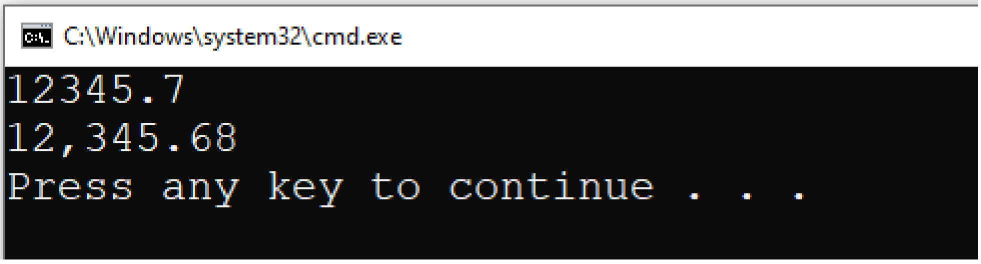
  Write a statement that uses System.out.printf to display the number as 12345.7.



```
num1 is 12345.7
Press any key to continue . . .
```

- Assume the double variable num2 contains the value 12345.6789.

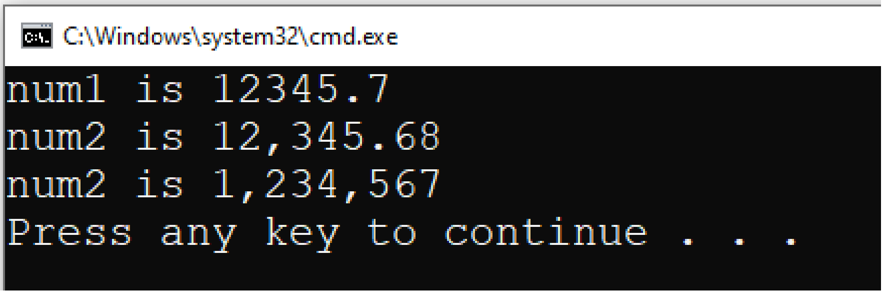
  Write an additional statement that uses System.out.printf to display num2 as 12,345.68.



```
12345.7
12,345.68
Press any key to continue . . .
```

- Assume the int variable num3 contains the value 1234567.

  Write a statement that uses System.out.printf to display the number as 1,234,567.



```
num1 is 12345.7
num2 is 12,345.68
num2 is 1,234,567
Press any key to continue . . .
```

Exercise 5

Goal: Create a program in Java, using printf to format the output as required.

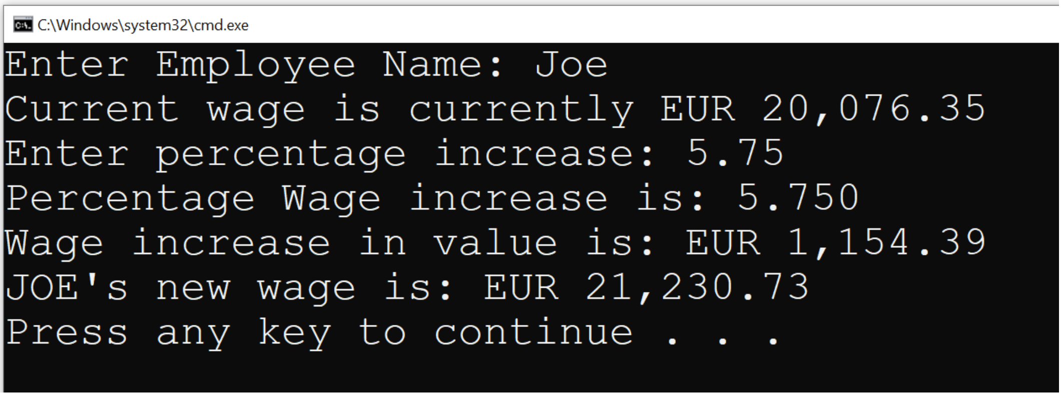
Create a new a Java program called JavaPrintf5, that will calculate a wage increase when a percentage increase is specified. The program should have a wage variable containing the value 20076.345. It should then output the new result similar to as shown below.

Prompt should be used for the following:

- Enter Employee name
- Enter Percentage wage increase

Ensure that the output of the wage shows the wage with commas separating the thousands, has 2 decimal places, and the percentage is shown with 3 decimal places.

The employee name should also appear as all caps in the final line of output (all using printf).

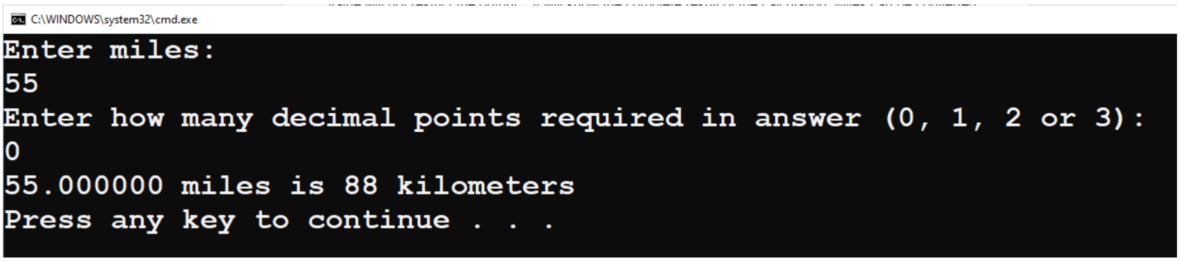


```
Enter Employee Name: Joe
Current wage is currently EUR 20,076.35
Enter percentage increase: 5.75
Percentage Wage increase is: 5.750
Wage increase in value is: EUR 1,154.39
JOE's new wage is: EUR 21,230.73
Press any key to continue . . .
```

Exercise 6

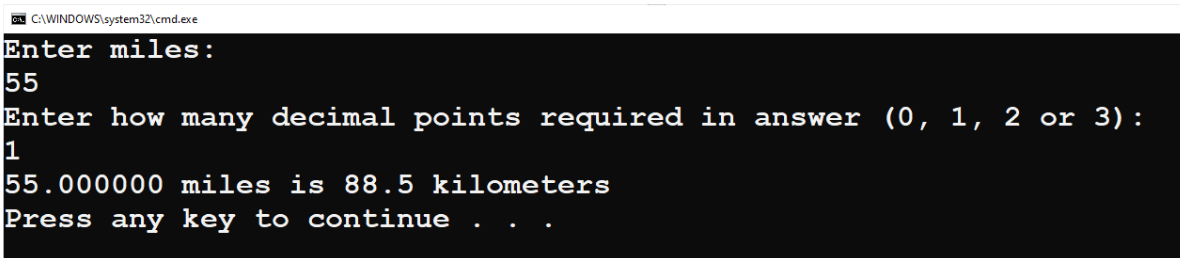
Goal: Create a program in Java, using printf to format the output as required.

Create a new a Java program called JavaConvertKms that will convert from miles to kilometres. Your program should allow the user to enter how many numbers after the decimal point will appear in the answer, allowing them to input 0, 1, 2 or 3, and then outputting the appropriate answer. Inputting any other value will not restrict the output – it will show the complete result of the calculation. Miles can be converted to kilometres by multiplying by 1.609. Sample output is shown below:



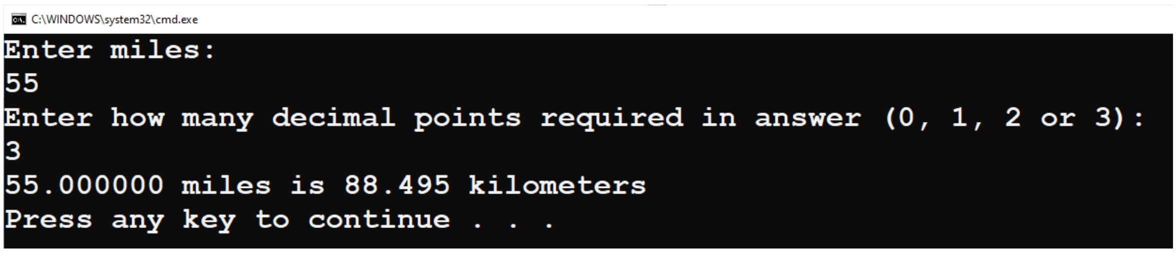
```
Enter miles:
55
Enter how many decimal points required in answer (0, 1, 2 or 3):
0
55.000000 miles is 88 kilometers
Press any key to continue . . .
```

or



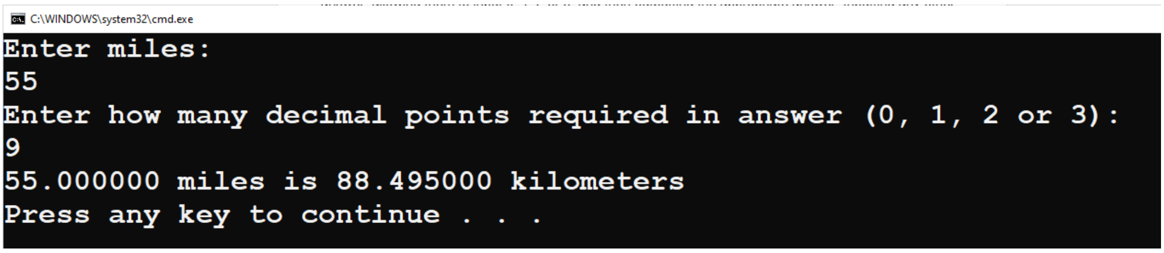
```
Enter miles:
55
Enter how many decimal points required in answer (0, 1, 2 or 3):
1
55.000000 miles is 88.5 kilometers
Press any key to continue . . .
```

or



```
Enter miles:
55
Enter how many decimal points required in answer (0, 1, 2 or 3):
3
55.000000 miles is 88.495 kilometers
Press any key to continue . . .
```

or

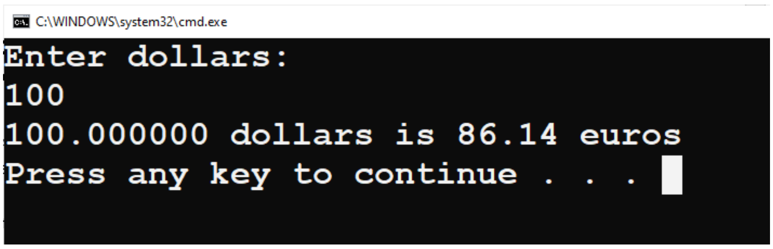


```
Enter miles:
55
Enter how many decimal points required in answer (0, 1, 2 or 3):
9
55.000000 miles is 88.495000 kilometers
Press any key to continue . . .
```

Exercise 7

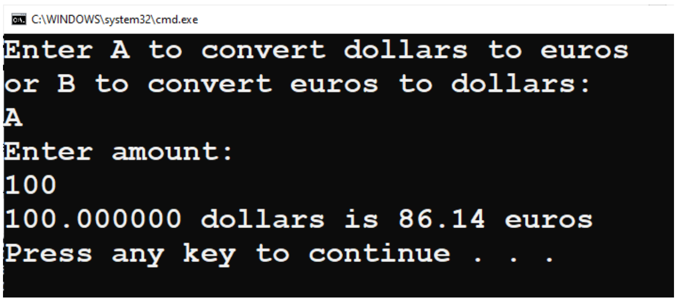
Goal: Create a program in Java, using printf to format the output as required.

Create a new a Java program called JavaConvertCurrency that will convert US dollars to Euros. Your program should allow the user to enter a dollar amount, and then output the equivalent in euros, to 2 decimal places.
Assume that each dollar buys you 0.86143522 euros.



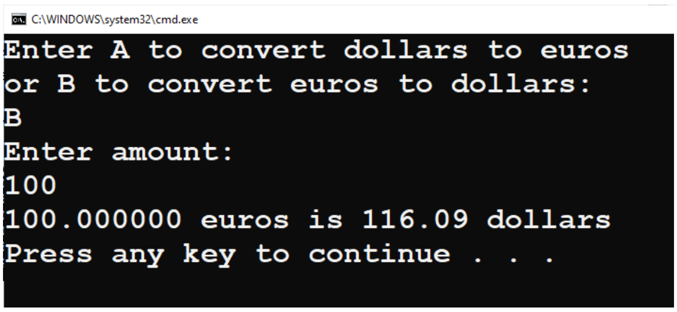
```
Enter dollars:
100
100.000000 dollars is 86.14 euros
Press any key to continue . . .
```

Amend your program so that the user **can choose the conversion** – from dollars to euros, or from euros to dollars, eg:



```
Enter A to convert dollars to euros
or B to convert euros to dollars:
A
Enter amount:
100
100.000000 dollars is 86.14 euros
Press any key to continue . . .
```

Or



```
Enter A to convert dollars to euros
or B to convert euros to dollars:
B
Enter amount:
100
100.000000 euros is 116.09 dollars
Press any key to continue . . .
```

Write this program in 2 different ways (you can save as 2 different files):

- Using an if statement
- Using a switch statement

Exercise 8

Goal: The goal of this exercise is to create a Java program that generates a simple report for an online shopping website. The report should provide details of an order, including the type of product purchased and the total price. Utilize a `switch` statement to identify the type of product and use `printf` to display the report in a formatted manner.

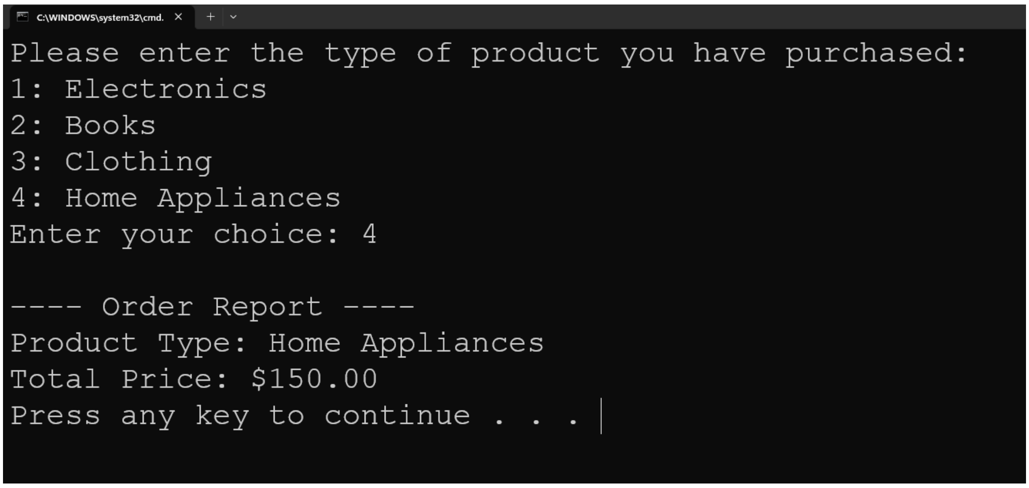
Create a new file called `OnlineShoppingReportGenerator` for this exercise.

Instructions:

1. **Create a Java Class:** Initialize a new Java class named `OnlineShoppingReportGenerator`.
2. **Define the Main Method:** Include the `main` method in your class.
3. **Implement User Input for Product Type:** Prompt the user to enter the type of product they have purchased. The options should be:
   - `1`: Electronics
   - `2`: Books
   - `3`: Clothing
   - `4`: Home Appliances

   Use a `switch` statement to identify the product type and calculate the total price based on user input. The total price is simply a placeholder and can be hard-coded.

4. **Use printf:** Utilize the `printf` method to display the report. The report should indicate the type of product purchased and its total price in a structured format.



```
Please enter the type of product you have purchased:
1: Electronics
2: Books
3: Clothing
4: Home Appliances
Enter your choice: 4

---- Order Report ----
Product Type: Home Appliances
Total Price: $150.00
Press any key to continue . . .
```

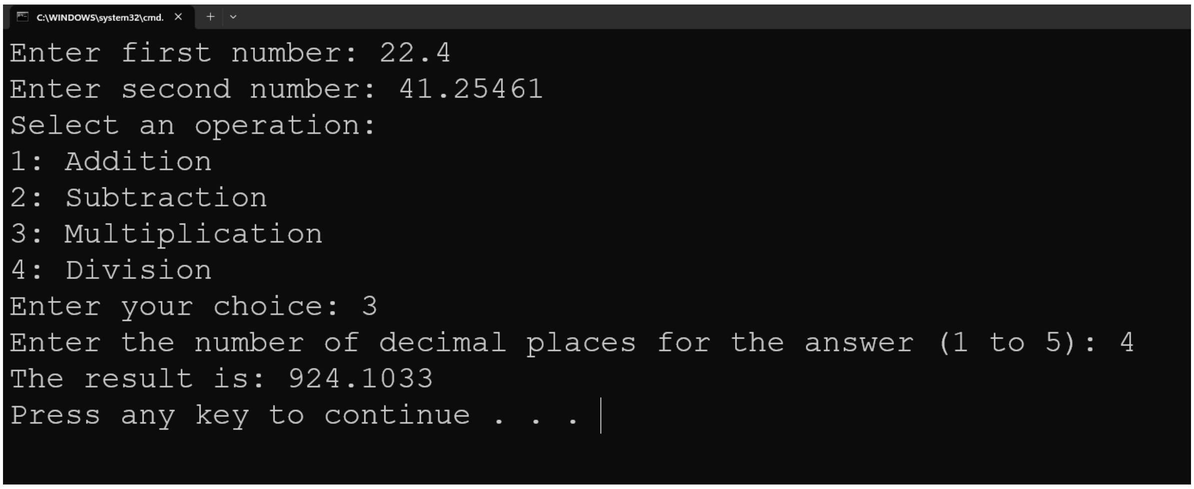
Exercise 9

Goal: The goal of this exercise is to create a Java program that functions as a simple calculator. The calculator should allow the user to perform basic arithmetic operations and specify the number of decimal places in the answer output, ranging from 1 to 5. Use switch statements for this exercise, do not use any if statements.

Create a new file called `DecimalPrecisionCalculator` for this exercise.

Instructions:

1. **Create a Java Class:** Initialize a new Java class named `DecimalPrecisionCalculator`.
2. **Define the Main Method:** Include the `main` method in your class.
3. **Implement User Input for Arithmetic Operation:** Prompt the user to select an arithmetic operation (Addition, Subtraction, Multiplication, Division).
4. **Implement User Input for Decimal Precision:** Prompt the user to specify the number of decimal places for the output. The user should be able to choose between 1 and 5 decimal places.
5. **Perform Calculation:** Based on the user's input, perform the calculation and display the result with the specified number of decimal places.



```
Enter first number: 22.4
Enter second number: 41.25461
Select an operation:
1: Addition
2: Subtraction
3: Multiplication
4: Division
Enter your choice: 3
Enter the number of decimal places for the answer (1 to 5): 4
The result is: 924.1033
Press any key to continue . . .
```

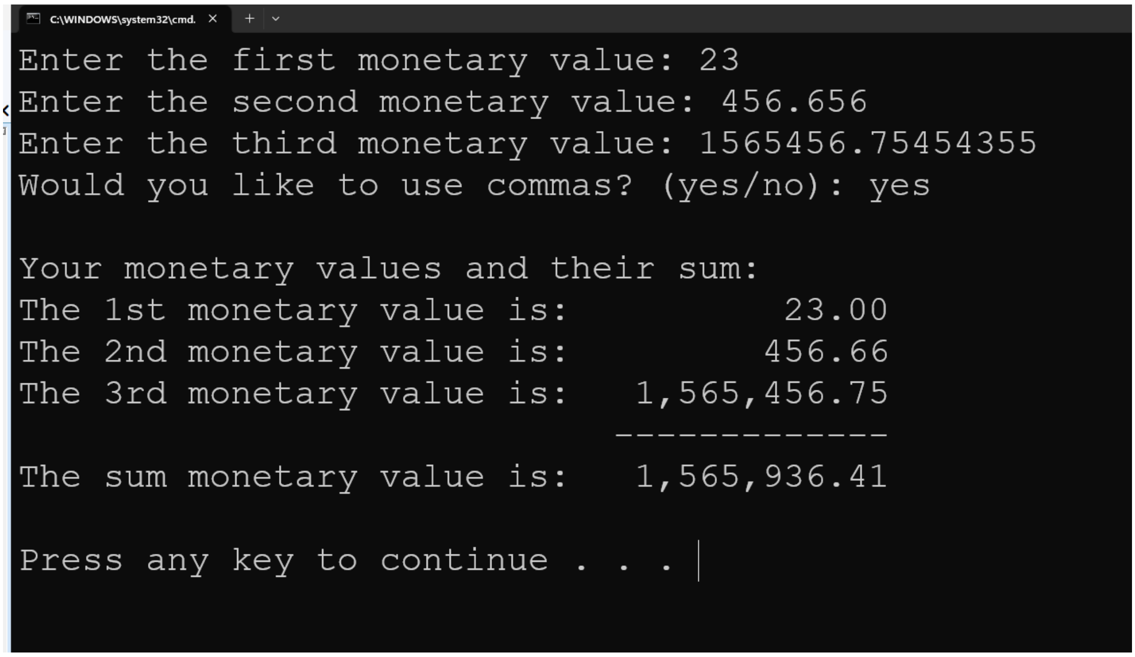
Exercise 10

Goal: The goal of this exercise is to create a Java program that allows the user to enter three monetary values. The program will then display the sum of these values, formatted according to user specifications for comma usage. The output will always present two only decimal places.

Create a new file called `MonetarySummation` for this exercise.

Instructions:

1. **Create a Java Class:** Initialize a new Java class named `MonetarySummation`.
2. **Define the Main Method:** Include the `main` method in your class.
3. **User Input for Monetary Values:** Prompt the user to enter three monetary values.
4. **User Input for Comma Usage:** Prompt the user to indicate whether commas should be used in the output.
5. **Perform Summation:** Add the three monetary values.
6. **Output:** Use `printf` to display the sum with two decimal places, and format it according to the user's specifications for leading zeros and comma usage.



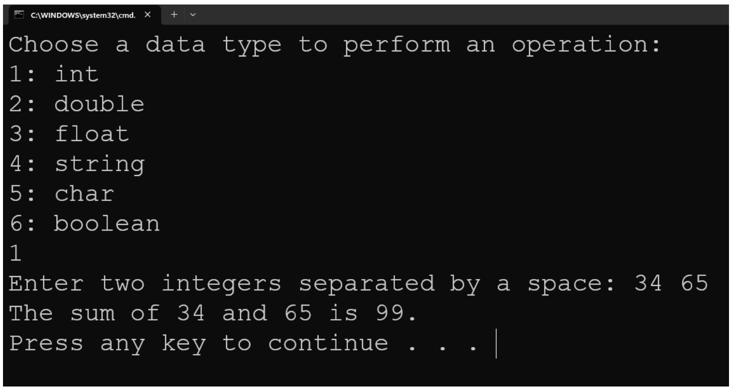
```
Enter the first monetary value: 23
Enter the second monetary value: 456.656
Enter the third monetary value: 1565456.75454355
Would you like to use commas? (yes/no): yes

Your monetary values and their sum:
The 1st monetary value is:         23.00
The 2nd monetary value is:        456.66
The 3rd monetary value is:   1,565,456.75
                            -------------
The sum monetary value is:   1,565,936.41

Press any key to continue . . .
```

Exercise 11

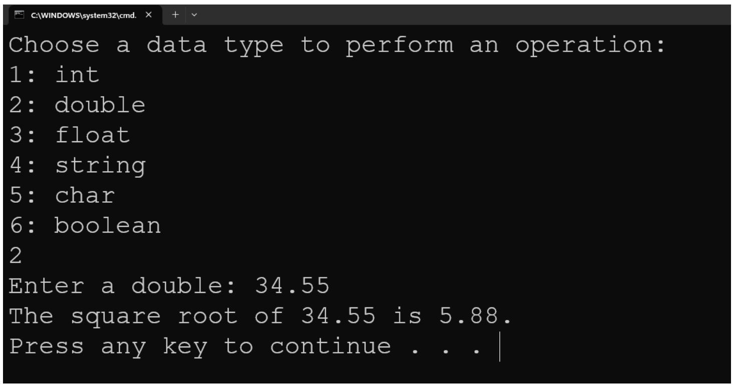
Goal: To create a program in Java that demonstrates usage of various data types (int, double, float, string, char, boolean) and employs switch statements for decision-making. The program will prompt the user to select a data type and then proceed to perform an operation associated with that data type. All outputs should be formatted using `printf`.

Instructions:

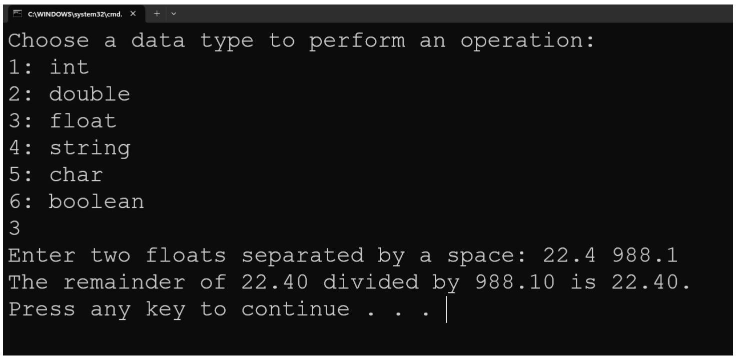
1. Create a new file called `MultiTypeSwitcher.java` for this exercise.
2. Use a `Scanner` object to allow the user to input their choice.
3. Show the user a menu to choose a data type from the following: int, double, float, string, char, boolean.
4. After the user makes a selection, perform an operation relevant to that data type. Below are the operations for each type:
   - **int**: Add two integers.
   - **double**: Calculate the square root of a double.
   - **float**: Compute the remainder when one float is divided by another.
   - **string**: Concatenate two strings.
   - **char**: Convert a lowercase char to uppercase.
   - **boolean**: Evaluate the logical `AND` of two boolean values.
5. Output the result of the operation using `printf`.
6. The program should not use any `if` statements, only `switch` statements for decision-making.



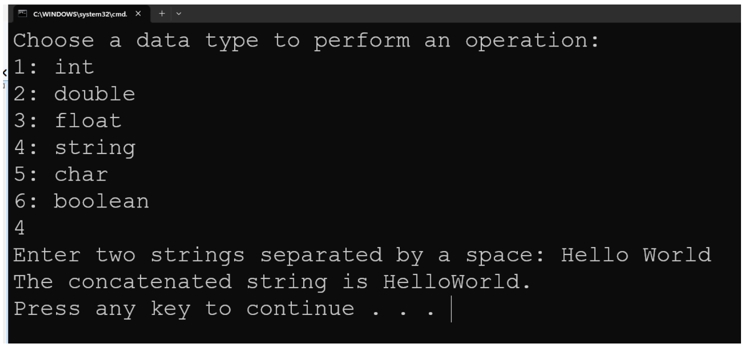
```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
1
Enter two integers separated by a space: 34 65
The sum of 34 and 65 is 99.
Press any key to continue . . .
```



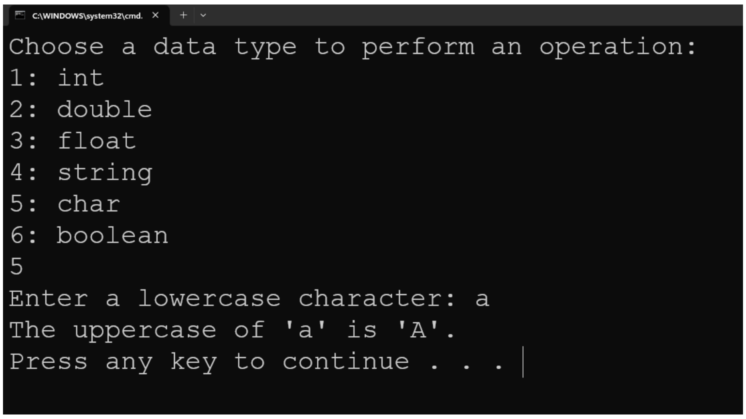
```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
2
Enter a double: 34.55
The square root of 34.55 is 5.88.
Press any key to continue . . .
```



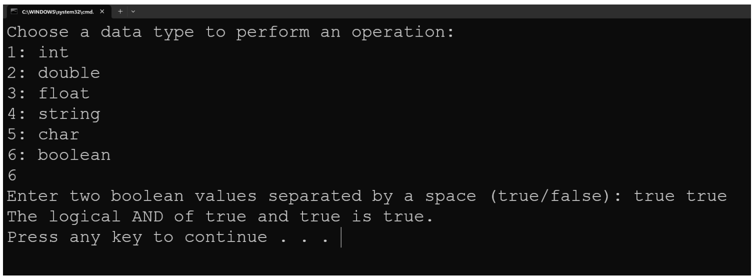
```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
3
Enter two floats separated by a space: 22.4 988.1
The remainder of 22.40 divided by 988.10 is 22.40.
Press any key to continue . . .
```



```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
4
Enter two strings separated by a space: Hello World
The concatenated string is HelloWorld.
Press any key to continue . . .
```



```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
5
Enter a lowercase character: a
The uppercase of 'a' is 'A'.
Press any key to continue . . .
```



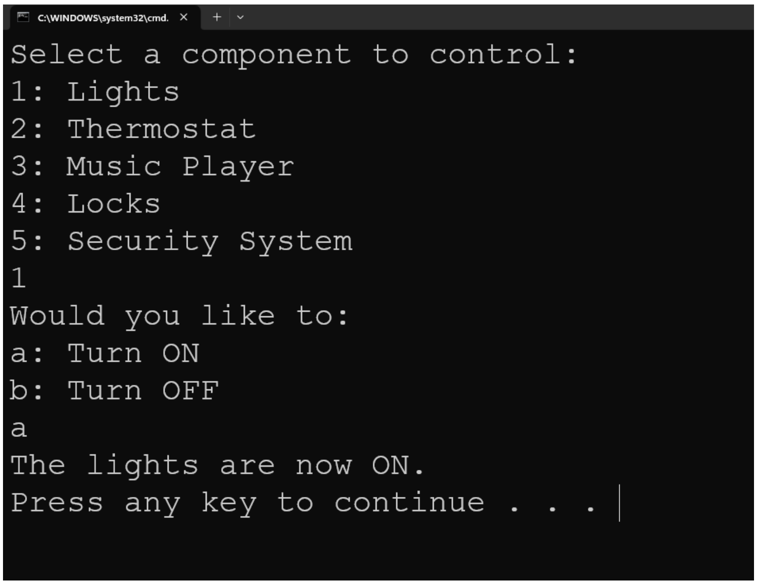
```
Choose a data type to perform an operation:
1: int
2: double
3: float
4: string
5: char
6: boolean
6
Enter two boolean values separated by a space (true/false): true true
The logical AND of true and true is true.
Press any key to continue . . .
```

Exercise 12

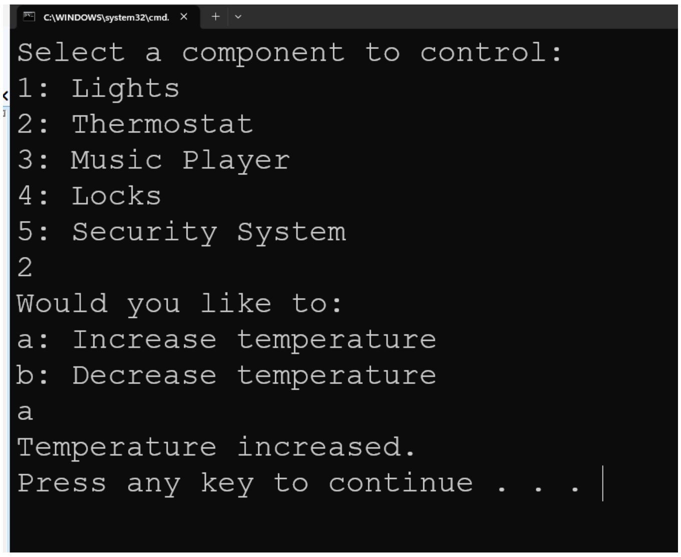
Goal: To create a program in Java that further exercises the use of various data types (int, double, float, string, char, boolean) while making decisions solely through switch statements. The program will simulate a user's interaction with different components of a smart home automation system. All outputs should be formatted using `printf`.

Instructions:

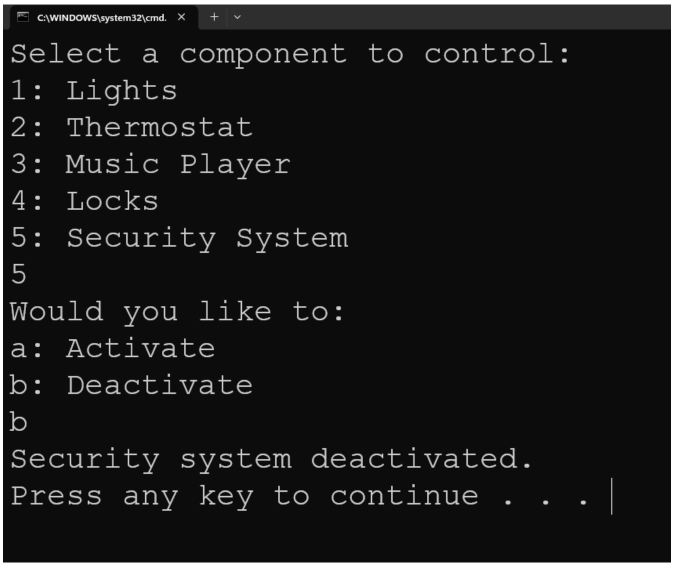
1. Create a new Java file named `SmartHomeSwitcher.java`.
2. Use a `Scanner` object to take input from the user.
3. Display a menu asking the user to choose a component of a smart home to control:
   - **1: Lights**
   - **2: Thermostat**
   - **3: Music Player**
   - **4: Locks**
   - **5: Security System**
4. After the user has made a selection, prompt them for an action specific to the chosen component. The operations are as follows:
   - **Lights**: Turn ON or OFF.
   - **Thermostat**: Increase or decrease temperature.
   - **Music Player**: Play or Pause.
   - **Locks**: Lock or Unlock.
   - **Security System**: Activate or Deactivate.
5. Use `printf` to display the resulting action performed on the selected component.
6. The program should not use any `if` statements; employ `switch` statements for all decision-making.



```
Select a component to control:
1: Lights
2: Thermostat
3: Music Player
4: Locks
5: Security System
1
Would you like to:
a: Turn ON
b: Turn OFF
a
The lights are now ON.
Press any key to continue . . .
```



```
Select a component to control:
1: Lights
2: Thermostat
3: Music Player
4: Locks
5: Security System
2
Would you like to:
a: Increase temperature
b: Decrease temperature
a
Temperature increased.
Press any key to continue . . .
```



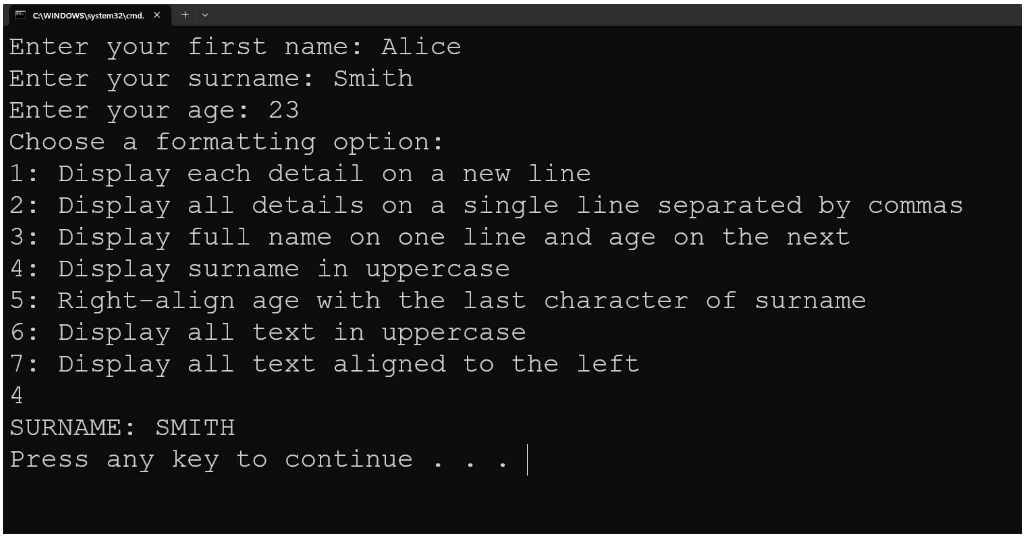
```
Select a component to control:
1: Lights
2: Thermostat
3: Music Player
4: Locks
5: Security System
5
Would you like to:
a: Activate
b: Deactivate
b
Security system deactivated.
Press any key to continue . . .
```

Exercise 13

Goal: To practice using `printf` for formatting output in Java.

Create a new Java file named `PersonalDetailsFormatter` for this exercise. The program should perform the following tasks:

1. Ask the user to enter their first name, surname, and age.
2. Provide a menu of different formatting options for how these details will be displayed on the console. The options should include:
   - Displaying the first name, surname, and age each on a new line.
   - Displaying them in a single line separated by commas.
   - Displaying the full name (first name and surname) on one line and age on the next line.
   - Displaying only the surname in uppercase letters.
   - Displaying all text in uppercase.
   - Displaying all text aligned to the left.
3. Include options that demonstrate different capabilities of `printf`, such as setting a minimum field width, left-justifying text, etc.
4. Ask the user to choose one of these formatting options.
5. Display the user's details as per the selected option using `printf`.
6. Only use `switch` statements for decision-making. Do not use any `if` statements.



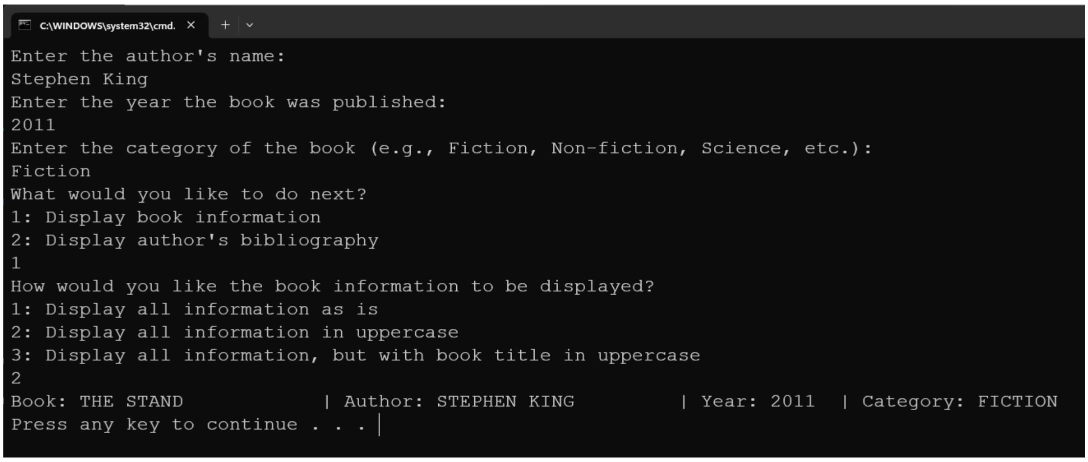
```
Enter your first name: Alice
Enter your surname: Smith
Enter your age: 23
Choose a formatting option:
1: Display each detail on a new line
2: Display all details on a single line separated by commas
3: Display full name on one line and age on the next
4: Display surname in uppercase
5: Right-align age with the last character of surname
6: Display all text in uppercase
7: Display all text aligned to the left
4
SURNAME: SMITH
Press any key to continue . . .
```

Exercise 14

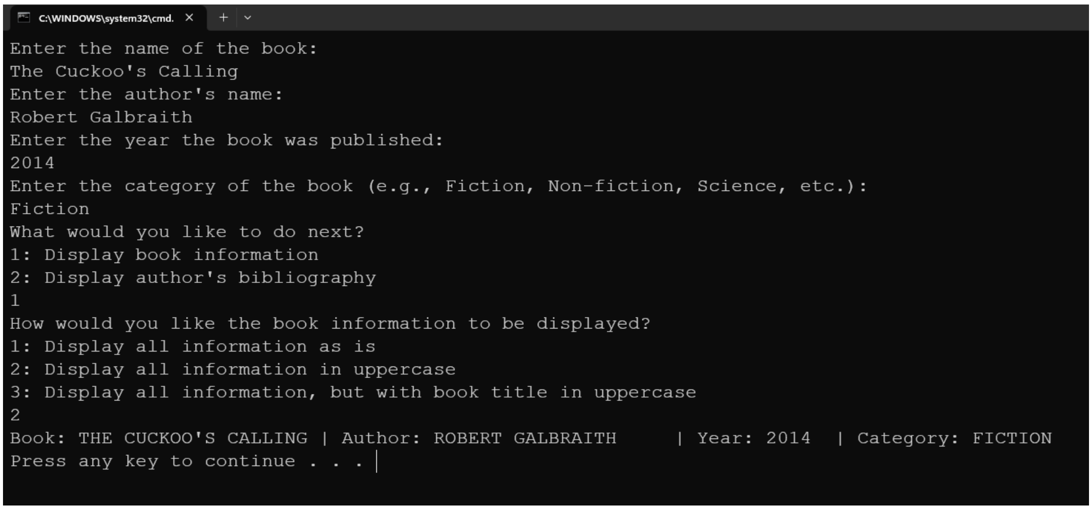
Goal: Create a program using `switch` statements and formatted output with `printf`

Create a new Java file named `LibraryManagement` for this exercise.

1. Write a program that simulates a basic library management system. Prompt the user to enter:
   - The name of a book
   - The author's name
   - The year the book was published
   - The category of the book (e.g., Fiction, Non-fiction, Science, etc.)
2. Ask the user what they would like to do next:
   - 1: Display book information
   - 2: Display author's bibliography
3. Use the first `switch` statement to handle the user's decision for displaying information.
4. If the user chooses to display the book information, ask them how they want it displayed:
   - 1: Display all information as is
   - 2: Display all information in uppercase
   - 3: Display all information, but with book title in uppercase
5. Use the second `switch` statement to format the book information output based on the user's choice.
6. Utilize `printf` extensively to ensure that the output is formatted neatly, aligning text and numbers appropriately. Use precision and padding functionalities of `printf` where necessary.



```
Enter the author's name:
Stephen King
Enter the year the book was published:
2011
Enter the category of the book (e.g., Fiction, Non-fiction, Science, etc.):
Fiction
What would you like to do next?
1: Display book information
2: Display author's bibliography
1
How would you like the book information to be displayed?
1: Display all information as is
2: Display all information in uppercase
3: Display all information, but with book title in uppercase
2
Book: THE STAND            | Author: STEPHEN KING         | Year: 2011  | Category: FICTION
Press any key to continue . . .
```



```
Enter the name of the book:
The Cuckoo's Calling
Enter the author's name:
Robert Galbraith
Enter the year the book was published:
2014
Enter the category of the book (e.g., Fiction, Non-fiction, Science, etc.):
Fiction
What would you like to do next?
1: Display book information
2: Display author's bibliography
1
How would you like the book information to be displayed?
1: Display all information as is
2: Display all information in uppercase
3: Display all information, but with book title in uppercase
2
Book: THE CUCKOO'S CALLING | Author: ROBERT GALBRAITH     | Year: 2014  | Category: FICTION
Press any key to continue . . .
```