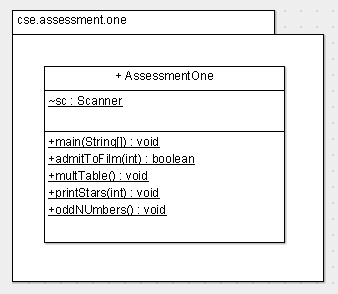
**CSE Java Fundamentals 2021**

**Assessment 1 Instructions**

**Weighting 40%**

**Time allowed: 2 hours**

**Open-book assessment.**



Notes:

* marks are deducted for compiler errors, logic errors and failure to follow the UML diagram
* insert your name at the top of the AssessmentOne.java file
* upload AssessmentOne.java to Moodle at the end
* code should be well formatted i.e. indented properly, proper placement of curly brackets, good variable names etc..
* use constants where appropriate e.g. CEILING is defined as 4 in *multTable()*
* there are plenty of printscreens of running programs as aids, please pay close attention to them
* assuming your code compiles, please leave main() uncommented, so that I can run it straight away

1. Cinema Admittance : you are asked to determine if, given a certain age whether a person is allowed to watch the film.
   1. in *main()*, ask the user to enter their *age* (use Scanner to get in this *int*).
   2. Call the method *admitToFilm()* passing down the variable *age* to the method. The *admitToFilm()* method determines whether a user can/cannot watch a film. Using its return value, output (**in *main()*** ) either: “Enjoy the film!” or “Sorry, you cannot watch this film.”.
   3. *admitToFilm()* method.
      1. the film in question is **15A** i.e. if one is aged 15 or older you can watch the film
      2. however, if you are younger than 15 you must have an adult with you before you are allowed to watch the film. Use the Scanner method *nextBoolean()* (true/false) to solve this part.

**[12 Marks]**

**Sample Output**



Figure 1



Figure 2



Figure 3

1. Multiplication table – you are asked to output the multiplication table for a user-inputted number.
   1. in *main()* call the *multTable()* method.
   2. in the *multTable()* method:
      1. ask the user for the value of their *int* value (call the variable *n*)
      2. using **both** *for* and *while* loops (which are **not** nested), output the multiplication table up to CELILING (a constant set to 4); for example, if 3 is entered by the user then output 3\*1, 3\*2, 3\*3 and 3\*4 (where 4 is the max CEILING value).

**[10 Marks]**

**Sample Output**

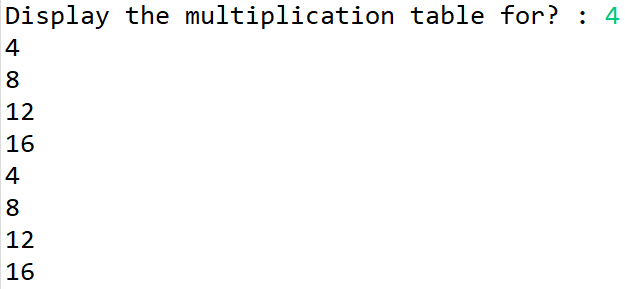
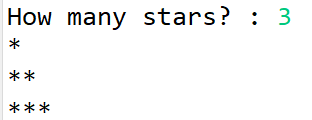


Figure 4

1. Printing stars – you are asked to output a particular star pattern.
   1. in *main()* ask the user “How many stars?”. Store the user input in an *int* variable named ‘*numStars’*. Invoke the method ‘*printStars()*’ passing down ‘*numStars’*.
   2. in *printStars()* do the following:
      1. using nested ‘*for’* loops output the pattern as follows (assuming *numStars* is 3):  
         
      2. Using a second set of nested ‘*for’* loops output the pattern as follows (again, assuming *numStars* is 3):  
         

**[12 Marks]**

**Sample Output**

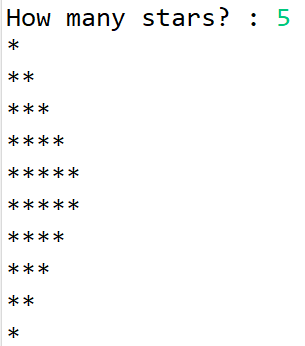


Figure 5

1. Given a range (specified by the user), print out the odd numbers in that range.
   1. In a method *oddNumbers()*, ask the user for a low number and a high number – store in 2 *int* variables, ‘*low’* and ‘*high’* respectively. Using a *for* loop and the ***continue*** keyword, output only the odd numbers in the range. Note that if **either or both** the low and high numbers are odd, they **must** be included in the output (see sample output). Note also that ***continue*** MUST cause the *System.out.println()* to be skipped.

**[10 Marks]**

**Sample Output**

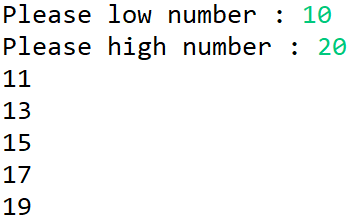
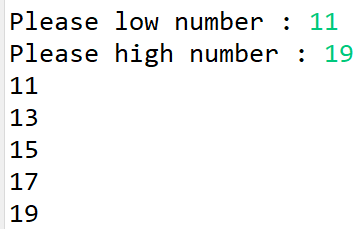
 

Figure 6 Figure 7

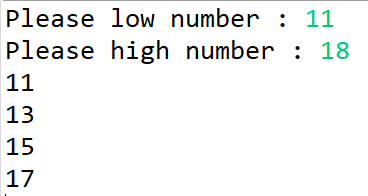
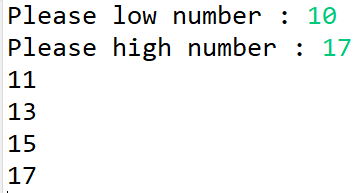
 

Figure 8 Figure 9