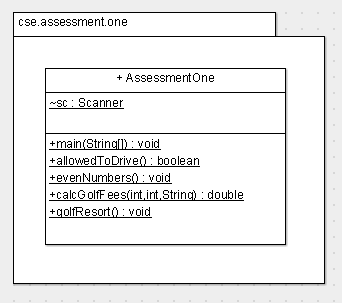
**CSE Java Fundamentals 2022**

**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 (in a comment)
* 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. ADULT\_FEE instead of 50
* there are plenty of printscreens of running programs as aids, please pay close attention to them

1. Allowed to drive the Car : you are asked to determine if, based on answers to certain questions, whether a person is allowed to drive a car.
   1. in *main()*, call the method *allowedToDrive()*. The *allowedToDrive()* method determines whether a person is/is not allowed to drive the car. Using its return value, output (in *main()* ) either: “Enjoy driving your car!” or “Sorry, you cannot drive the car.”.
   2. *allowedToDrive()* method.
      1. you are allowed to drive a car if you have a full drivers licence AND are insured AND the car is taxed. **Note** that the questions have ***boolean*** inputs from the user! (see Sample Output below).
      2. using **nested-if** statements, implement the logic; note that when using nested-if statements, once one question is *false*, there is no need to ask any other question i.e. if any of the questions are *false*, you cannot drive the car (see Sample Output below).

**[10 Marks]**

**Sample Output**

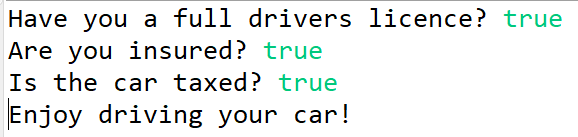


Figure 1 – the only way you can drive the car

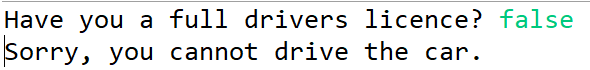


Figure 2 – first question is false (note: no other questions asked)

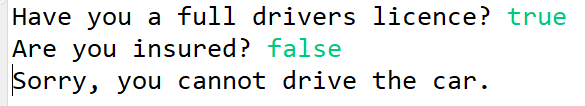


Figure 3 – second question is false (note: no other questions asked)

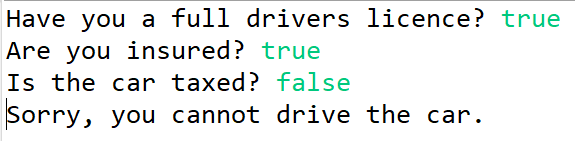


Figure 4 – third question is false

1. Given a range (specified by the user), print out the **even** numbers in that range.
   1. In a method *evenNumbers()*, ask the user for a low number and a high number – store in 2 *int* variables, ‘*low’* and ‘*high’* respectively. Using a *for* loop, output only the **even** numbers in the range. Note that if **either or both** the low and high numbers are **even**, they **must** be included in the output (see Sample Output).

**[10 Marks]**

**Sample Output**

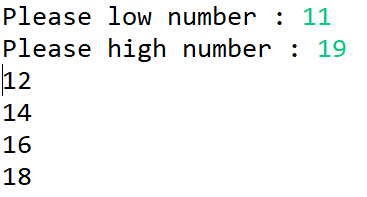
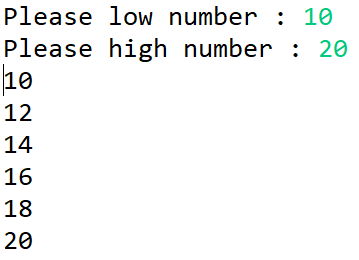
 

Figure 5 – both inputs odd Figure 6– both inputs even

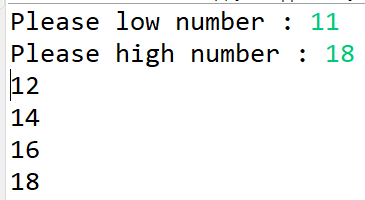
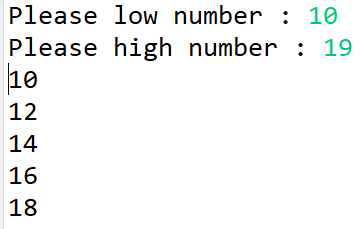
 

Figure 7 – low is odd, high is even Figure 8 - low is even, high is odd

1. Golf Resort : You are required to calculate the golf fees to be paid based on the following user input:

* the age of the golfer (which determines the initial fee). The following table applies. Note that if you are younger than 10 years of age, you are not allowed to play.

|  |  |  |
| --- | --- | --- |
| **Age** | **Category** | **Fee (per round of golf)** |
| 10-17 (inclusive) | Junior | €20 |
| 18-59(inclusive) | Adult | €50 |
| >=60 | Senior | €40 |

* the number of rounds you are paying for (assume rounds are per day) – you can book between 1 and 5 rounds (inclusive) only.
* the day of the week – you pay 10% extra on the weekend (Saturday or Sunday).

**[20 Marks]**

1. in *main()* call *golfResort()*
2. **the rest of the instructions are for the *golfResort()* method:**
3. ask the user for his/her *age*; use Scanner to get in this *int* from the keyboard
4. if he/she is too young, output this fact and do nothing else (use a constant instead of the literal integer 10 in your *if* statement)
5. if he/she is old enough to play:
   1. ask the user for the number of rounds of golfs they are paying for (an *int*); use Scanner to get this user input
   2. if the number of rounds is invalid e.g. -3, 0, 6, 7, -2 etc… output an error and do nothing else (again use constants for the lowest/highest valid values)
   3. if the number of rounds is valid:
      1. ask the use what day of the week they are playing; use Scanner to get in this *String* variable *dayOfWeek*; acceptable Strings are the first 3 letters for any day of the week e.g. “Mon” for Monday, “Tue” for Tuesday etc..
      2. convert *dayOfWeek* to uppercase (this will make the following *switch* statement cleaner)
      3. using a *switch* statement determine if *dayOfWeek* is indeed a valid day of the week i.e. the user could have typed in “Abc” for the day of the week.
         1. If *dayOfWeek* is valid, set a *boolean* *dayOk* to *true* (declare this variable at the top of *golfResort()* and initialise it to *false.*
         2. If *dayOfWeek* is invalid, flag an error.
      4. if *dayOfWeek* was valid (*dayOk* will tell you this), then :
         1. call the method *calcGolfFees* passing down the age, number of rounds and day of week. *calcGolfFees* will return the fees to be charged.
         2. Output the fees.
      5. if *dayOfWeek* was invalid, do nothing

*calcGolfFees()* method

1. declare constants for the senior, adult and junior fees
2. declare constants for the senior and adult age
3. using an *if-else if-else* structure calculate the basic fee
4. using a *switch* structure, apply a 10% additional charge if the day of the week is a Saturday or Sunday
5. return the fee calculated

**Sample Output**

**Errors in the input**

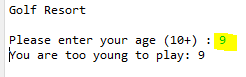


Figure 9

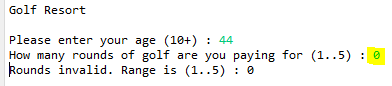


Figure 10

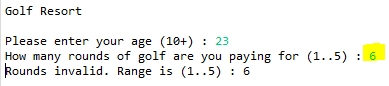


Figure 11

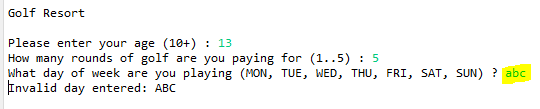


Figure 12

**Valid input:**

**Junior x 1 during the week:**

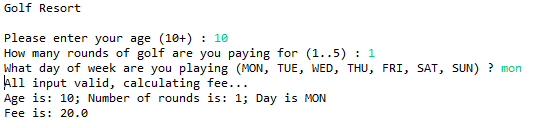


Figure 13

**Junior x 2 at the weekend:**

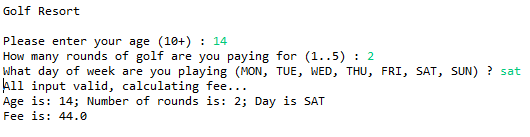


Figure 14

**Adult x 2 at the weekend:**

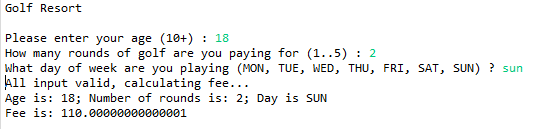


Figure 15

If you get 110.0 then that is absolutely fine.

**Senior x 3 at the weekend:**

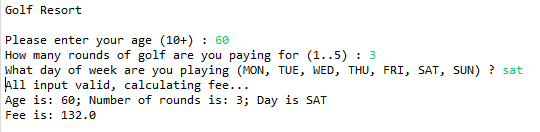


Figure 16