Assessment Item 1:

Basic Algorithms in Java

Bryce Sandilands: 11649401

ITC206-201860

Task One: Documentation

- First the `filingStatus` and `taxableIncome` are gathered from the user.
- `totalTaxAmount` is created and initialized to 0.
- The `switch` statement on `filingStatus` has cases to match the input from user. Each case calls the `dertermineTax` function which requires 6 arguments to be passed in, after a case has been matched the `break` statement is used to exit the `switch` statement.
- Following this the program displays the `totalTaxAmount` to the user and closes the Scanner object
- Lastly the `determineTax` method is used to compute the amount of tax based off the `taxableIncome` and the `upperBoundary` for each tax bracket. This method was designed to reduce code by using parameters instead of hard coded values for the `income` and `upperBoundary` values. The method works by checking if `income` is `<=` the `upperBoundary` for the tax bracket in ascending order. This eliminates redundant checking of both lower and `upperBoundary` values. Each conditional check finds the amount of money between each tax bracket, multiples it by the tax rate and adds them together until the income amount is reached where difference in income is found and multiplied by the tax rate, that then gets assigned to `tax` property.

Task One: Sample Outputs

```
Enter the filing status: 0
Enter the taxable income: 100000
Tax is: 21720.0
```

Run Succeeded | Time 447 ms

```
Enter the filing status: 1
Enter the taxable income: 85000
Tax is: 13625.0
```

Run Succeeded | Time 367 ms

```
Enter the filing status: 1
Enter the taxable income: 300339
Tax is: 76932.87
```

Run Succeeded | Time 367 ms

```
Enter the filing status: 3
Enter the taxable income: 525777
Tax is: 158381.95
```

Run Succeeded | Time 337 ms

- Output from Subject Outline
- New input data

Task Two: Documentation

- The `numberOfChildren, numberOfAdults, numberOfSeniors & totalCost` properties are created and initialized to 0.
- The user is then prompt to enter a group to add and the value is assigned to `enterNewGroup.`
- Following this the while loop condition `(enterNewGroup != 0)` is used to pretest if a new group needs to be added, if so the loop iterates over the number of entrants and assigns the values to `numberOfChildren, numberOfAdults, numberOfSeniors` accordingly
- Next the `numberOfChildren, numberOfAdults, numberOfSeniors` properties are passed into the `calculateTotalCost` method and the value is assigned to the `costOfGroup` property. The total entry charge is displayed to the user and the `costOfGroup` is then added to the accumulator property `totalCost`.
- The user is then prompt again to either enter a new group or finish the program.
- Much like Task One a separate function was designed to do the calculations of the total cost. This promotes the separations of concern principle.
- The method adds both the senior and adults together as `accompanyingAdults` and uses a ternary operator to find the number of `accompaniedChildren & unaccompaniedChildren.`
- It then multiples four categories of entrants by their ticket costs, adds the values together and assigns it to `total` which is then returned.

Task Two: Sample Outputs

```
Enter a group? (Yes = 1 / No = 0): 1
Enter the number of children (age 6–15): 5
Enter the number of adults (age 16–59): 2
Enter the number of seniors (age 60+): 1
Total Entry Charge is $44

Enter a group? (Yes = 1 / No = 0): 1
Enter the number of children (age 6–15): 1
Enter the number of adults (age 16–59): 2
Enter the number of seniors (age 60+): 0
Total Entry Charge is $22

Enter a group? (Yes = 1 / No = 0): 0
Total takings: $66
```

Run Succeeded | Time 353 ms

- Output from Subject Outline

```
Enter a group? (Yes = 1 / No = 0): 1
Enter the number of children (age 6–15): 9
Enter the number of adults (age 16–59): 3
Enter the number of seniors (age 60+): 2
Total Entry Charge is $76

Enter a group? (Yes = 1 / No = 0): 1
Enter the number of children (age 6–15): 2
Enter the number of adults (age 16–59): 5
Enter the number of seniors (age 60+): 11
Total Entry Charge is $142

Enter a group? (Yes = 1 / No = 0): 0
Total takings: $218
```

Run Succeeded | Time 476 ms

- New input data