

Department of Cyber Security  
Amrita School of Computing  
Amrita Vishwa Vidyapeetham, Chennai Campus  
Principals of Programming Languages

---

Subject Code: 20CYS312

Date:2024/12/20

Name: Sushant Yadav

Roll Number:CH.EN.U4CYS22067

---

Lab -IncomeTax

1. The task involves implementing a Haskell program to calculate income tax based on two tax regimes: the Old Tax Regime and the New Tax Regime. The user must input their income and select the tax regime to calculate the tax.

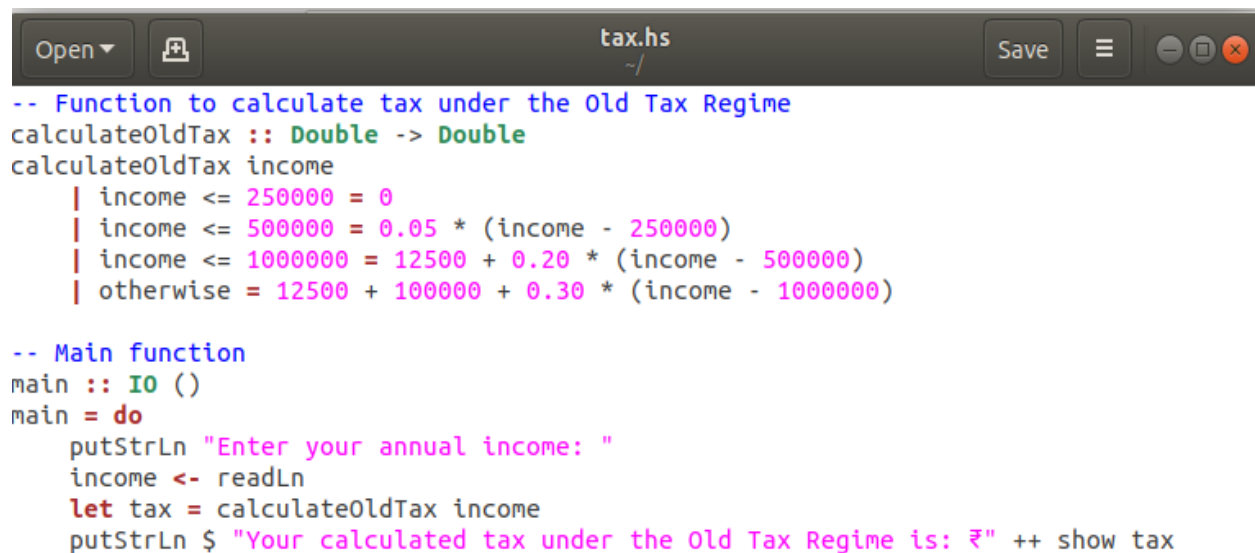
Old Tax Regime Up to ₹2.5 lakh: No tax

₹2.5 lakh to ₹5 lakh: 5%

₹5 lakh to ₹10 lakh: 20%

Above ₹10 lakh: 30%

Code:



```
-- Function to calculate tax under the Old Tax Regime
calculateOldTax :: Double -> Double
calculateOldTax income
    | income <= 250000 = 0
    | income <= 500000 = 0.05 * (income - 250000)
    | income <= 1000000 = 12500 + 0.20 * (income - 500000)
    | otherwise = 12500 + 100000 + 0.30 * (income - 1000000)

-- Main function
main :: IO ()
main = do
    putStrLn "Enter your annual income: "
    income <- readLn
    let tax = calculateOldTax income
    putStrLn $ "Your calculated tax under the Old Tax Regime is: ₹" ++ show tax
```

Explanation:

**1. Income Slabs:**

- If the income is up to ₹2.5 lakh, no tax is applied.
- Between ₹2.5 lakh and ₹5 lakh, 5% tax is applied on the amount above ₹2.5 lakh.
- Between ₹5 lakh and ₹10 lakh, 20% tax is applied on the amount above ₹5 lakh.
- Above ₹10 lakh, 30% tax is applied on the amount above ₹10 lakh.

**2. Computation Logic:**

- a. For each slab, the program calculates tax incrementally.
  - b. Pre-computed values are added for previous slabs to avoid recalculating.
3. **User Interaction:**
- a. The user inputs their annual income.
  - b. The program calculates and displays the total tax.

## Output:

```
asecomputerlab@hp-desktop: ~  
File Edit View Search Terminal Help  
asecomputerlab@hp-desktop:~$ gedit tax.hs  
^C  
asecomputerlab@hp-desktop:~$ runhaskell tax.hs  
Enter your annual income:  
40000  
Your calculated tax under the Old Tax Regime is: ₹0.0  
asecomputerlab@hp-desktop:~$ runhaskell tax.hs  
Enter your annual income:  
7000000  
Your calculated tax under the Old Tax Regime is: ₹1912500.0  
asecomputerlab@hp-desktop:~$
```

## Conclusion

The provided Haskell program effectively calculates income tax under the **Old Tax Regime** by adhering to the specified income tax slabs and rates. It incrementally applies the tax rates to the relevant portions of the income, ensuring accurate computation.

This implementation allows users to:

1. Input their annual income.
2. Receive an accurate tax calculation based on the given rules for the **Old Tax Regime**.

The program demonstrates the use of **conditional logic** (guards) to handle multiple income brackets and illustrates how **functional programming** in Haskell can be applied to real-world problems.

If required, this program can be expanded to include additional features, such as:

- Supporting the **New Tax Regime** for comparison.
- Allowing for deductions or exemptions.
- Improving user interactivity and validation.

2.The task involves implementing a Haskell program to calculate income tax based on two tax regimes: the Old Tax Regime and the New Tax Regime. The user must input their income and select the tax regime to calculate the tax.

New Tax Regime Up to ₹2.5 lakh: No tax

₹2.5 lakh to ₹5 lakh: 5%

₹5 lakh to ₹7.5 lakh: 10%

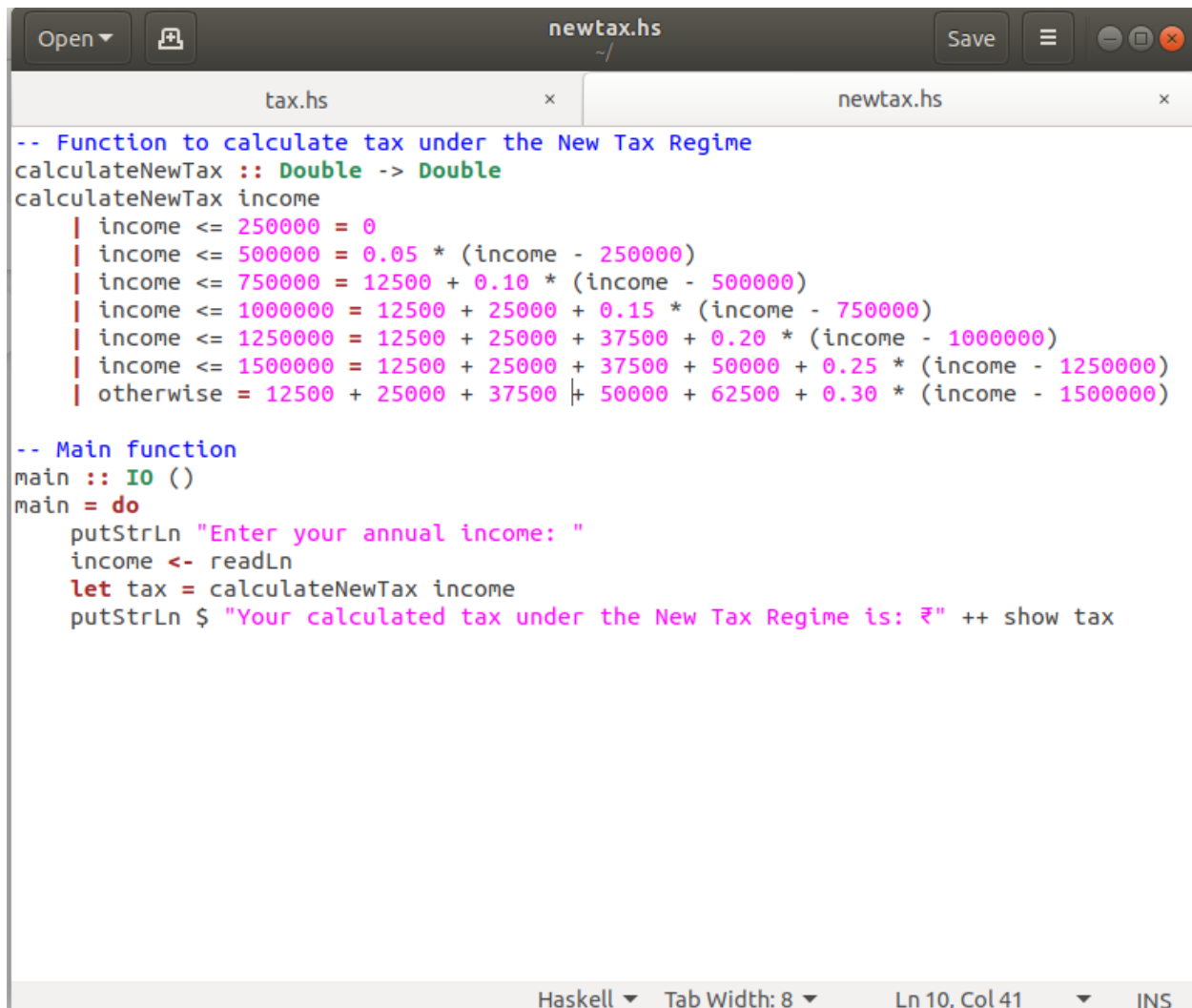
₹7.5 lakh to ₹10 lakh: 15%

₹10 lakh to ₹12.5 lakh: 20%

₹12.5 lakh to ₹15 lakh: 25%

Above ₹15 lakh: 30%

Code:

The image shows a screenshot of a Haskell code editor window titled 'newtax.hs'. The editor has a dark theme and shows two tabs: 'tax.hs' and 'newtax.hs'. The code is written in Haskell and defines a function 'calculateNewTax' and a 'main' function. The 'calculateNewTax' function uses guards to calculate tax based on income slabs. The 'main' function prompts the user for their annual income and prints the calculated tax. The status bar at the bottom indicates 'Haskell', 'Tab Width: 8', 'Ln 10, Col 41', and 'INS' mode.

```
-- Function to calculate tax under the New Tax Regime
calculateNewTax :: Double -> Double
calculateNewTax income
  | income <= 250000 = 0
  | income <= 500000 = 0.05 * (income - 250000)
  | income <= 750000 = 12500 + 0.10 * (income - 500000)
  | income <= 1000000 = 12500 + 25000 + 0.15 * (income - 750000)
  | income <= 1250000 = 12500 + 25000 + 37500 + 0.20 * (income - 1000000)
  | income <= 1500000 = 12500 + 25000 + 37500 + 50000 + 0.25 * (income - 1250000)
  | otherwise = 12500 + 25000 + 37500 + 50000 + 62500 + 0.30 * (income - 1500000)

-- Main function
main :: IO ()
main = do
  putStrLn "Enter your annual income: "
  income <- readLn
  let tax = calculateNewTax income
  putStrLn $ "Your calculated tax under the New Tax Regime is: ₹" ++ show tax
```

Explanation of the Code:

1. **Income Tax Slabs:**

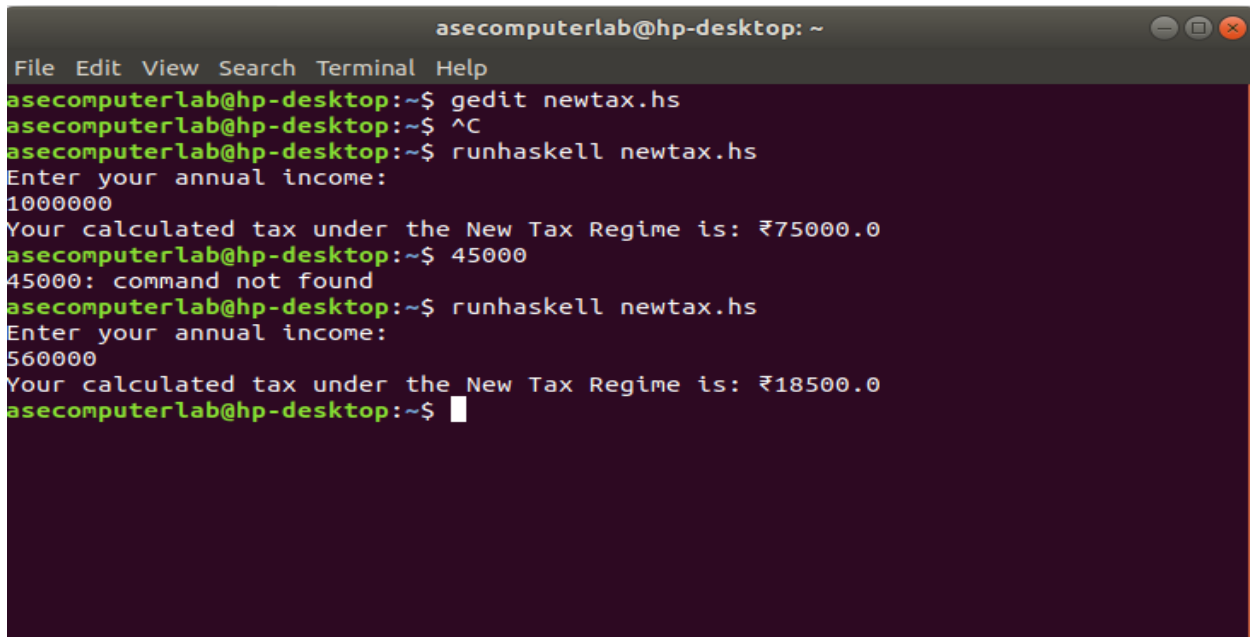
- ₹2.5 lakh and below:** No tax.
- ₹2.5 lakh to ₹5 lakh:** 5% tax on the portion above ₹2.5 lakh.
- ₹5 lakh to ₹7.5 lakh:** 10% tax on the portion above ₹5 lakh.
- ₹7.5 lakh to ₹10 lakh:** 15% tax on the portion above ₹7.5 lakh.
- ₹10 lakh to ₹12.5 lakh:** 20% tax on the portion above ₹10 lakh.
- ₹12.5 lakh to ₹15 lakh:** 25% tax on the portion above ₹12.5 lakh.
- Above ₹15 lakh:** 30% tax on the portion above ₹15 lakh.

2. **Logic:**

- The function `calculateNewTax` uses **guards** to compute tax incrementally for each slab.

- b. Pre-computed tax values for the lower slabs are added in sequence to simplify calculations.
3. **User Interaction:**
  - a. The program prompts the user to enter their annual income.
  - b. It calculates and displays the total tax based on the **New Tax Regime**.

Output:

A terminal window titled 'asecomputerlab@hp-desktop: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
asecomputerlab@hp-desktop:~$ gedit newtax.hs
asecomputerlab@hp-desktop:~$ ^C
asecomputerlab@hp-desktop:~$ runhaskell newtax.hs
Enter your annual income:
1000000
Your calculated tax under the New Tax Regime is: ₹75000.0
asecomputerlab@hp-desktop:~$ 45000
45000: command not found
asecomputerlab@hp-desktop:~$ runhaskell newtax.hs
Enter your annual income:
560000
Your calculated tax under the New Tax Regime is: ₹18500.0
asecomputerlab@hp-desktop:~$
```

## Conclusion:

The Haskell program successfully calculates income tax under the **New Tax Regime** by implementing the specified income slabs and tax rates. The program:

1. Handles multiple tax slabs and calculates the tax incrementally for each slab.
2. Uses clear and efficient logic with **guards** to apply the appropriate tax rate based on the user's income.
3. Provides a simple and interactive way for the user to input their income and get an accurate tax calculation.

This implementation showcases how **functional programming** in Haskell can be applied to solve practical problems like tax computation.

### Key Features:

- The program adheres to the New Tax Regime, with tax rates ranging from 5% to 30% based on the income slabs.
- Incremental calculations ensure correctness and efficiency.

### Next Steps:

If needed, this program can be extended to:

- Include the **Old Tax Regime** for comparison.
- Add validation for incorrect inputs.
- Display a breakdown of the tax calculations for each slab.

Final combined haskell code for the income tax :

The task involves implementing a Haskell program to calculate income tax based on two tax regimes: the Old Tax Regime and the New Tax Regime. The user must input their income and select the tax regime to calculate the tax.

### Old Tax Regime

- **Up to ₹2.5 lakh:** No tax
- **₹2.5 lakh to ₹5 lakh:** 5%
- **₹5 lakh to ₹10 lakh:** 20%
- **Above ₹10 lakh:** 30%

### New Tax Regime

- **Up to ₹2.5 lakh:** No tax
- **₹2.5 lakh to ₹5 lakh:** 5%
- **₹5 lakh to ₹7.5 lakh:** 10%
- **₹7.5 lakh to ₹10 lakh:** 15%
- **₹10 lakh to ₹12.5 lakh:** 20%

- ₹12.5 lakh to ₹15 lakh: 25%
- Above ₹15 lakh: 30%

Here's a Haskell implementation of the program:

Code:

```

incometax.hs
Open Save
import Text.Printf (printf)

-- Function to calculate tax under the old regime
calculateOldTax :: Double -> Double
calculateOldTax income
    | income <= 250000 = 0
    | income <= 500000 = 0.05 * (income - 250000)
    | income <= 1000000 = 12500 + 0.20 * (income - 500000)
    | otherwise = 12500 + 100000 + 0.30 * (income - 1000000)

-- Function to calculate tax under the new regime
calculateNewTax :: Double -> Double
calculateNewTax income
    | income <= 250000 = 0
    | income <= 500000 = 0.05 * (income - 250000)
    | income <= 750000 = 12500 + 0.10 * (income - 500000)
    | income <= 1000000 = 12500 + 25000 + 0.15 * (income - 750000)
    | income <= 1250000 = 12500 + 25000 + 37500 + 0.20 * (income - 1000000)
    | income <= 1500000 = 12500 + 25000 + 37500 + 50000 + 0.25 * (income - 1250000)
    | otherwise = 12500 + 25000 + 37500 + 50000 + 62500 + 0.30 * (income - 1500000)

-- Main function
main :: IO ()
main = do
    putStrLn "Enter your annual income: "
    income <- readLn
    putStrLn "Choose the tax regime (1 for Old, 2 for New): "
    regime <- readLn
    let tax = if regime == 1 then calculateOldTax income else calculateNewTax income
    printf "Your calculated tax is: ₹%.2f\n" tax

```

## Explanation

### 1. Tax Calculation for Each Regime:

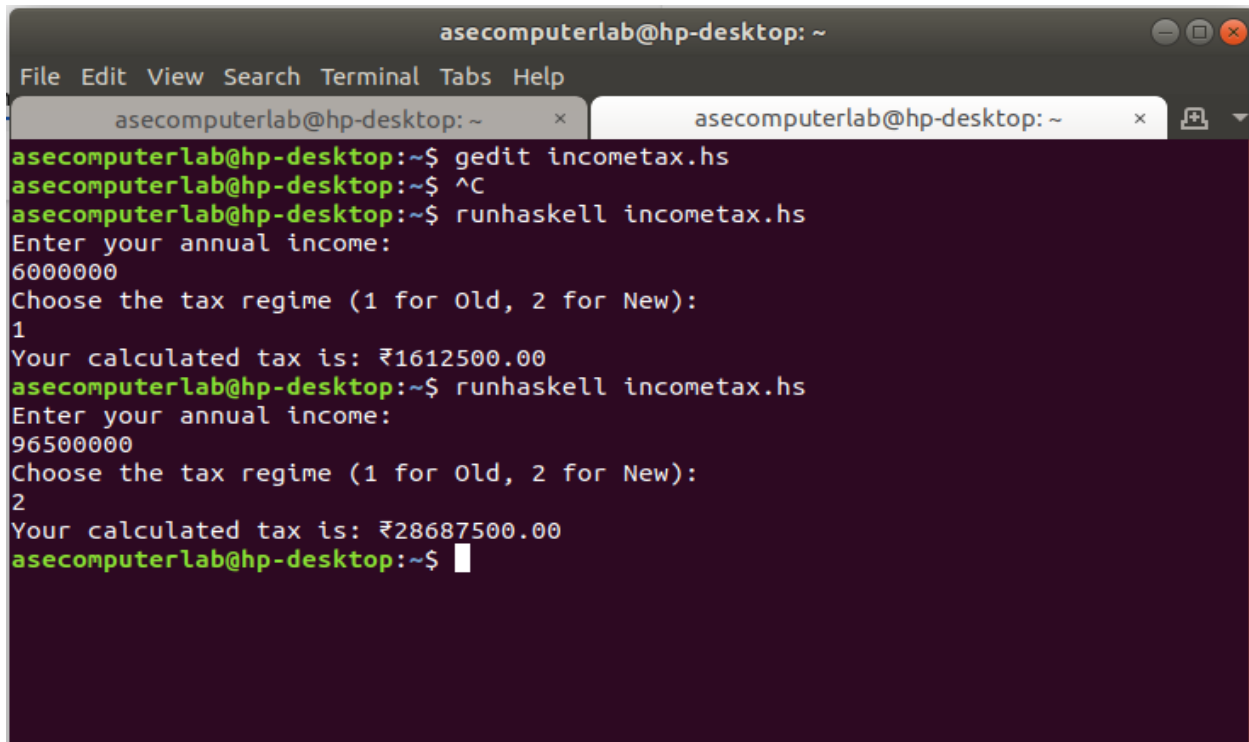
- Old Tax Regime:** The function `calculateOldTax` uses the given slabs and applies the corresponding rates incrementally. Pre-computed values for lower slabs are added to simplify calculations.
- New Tax Regime:** The function `calculateNewTax` similarly applies the slabs and rates as specified for the new system.

### 2. Input and User Interaction:



- a. The user is prompted to enter their **annual income**.
  - b. They choose between the **Old Tax Regime (1)** and **New Tax Regime (2)**.
3. **Conditional Logic:**
- a. Based on the user's choice, the appropriate tax function is called to compute the tax.
  - b. The calculated tax is displayed to the user.

Output:



```
asecomputerlab@hp-desktop: ~  
File Edit View Search Terminal Tabs Help  
asecomputerlab@hp-desktop: ~ x asecomputerlab@hp-desktop: ~ x  
asecomputerlab@hp-desktop:~$ gedit incometax.hs  
asecomputerlab@hp-desktop:~$ ^C  
asecomputerlab@hp-desktop:~$ runhaskell incometax.hs  
Enter your annual income:  
6000000  
Choose the tax regime (1 for Old, 2 for New):  
1  
Your calculated tax is: ₹1612500.00  
asecomputerlab@hp-desktop:~$ runhaskell incometax.hs  
Enter your annual income:  
96500000  
Choose the tax regime (1 for Old, 2 for New):  
2  
Your calculated tax is: ₹28687500.00  
asecomputerlab@hp-desktop:~$
```

## Conclusion

This Haskell program:

1. Accurately calculates income tax for both **Old** and **New Tax Regimes** by following the respective tax slabs and rates.
2. Demonstrates the use of functional programming constructs like **guards** and **pattern matching** for clarity and efficiency.
3. Offers an interactive interface for the user to input their income and select the tax regime.

## Highlights:

- The program adheres strictly to the tax rules, ensuring correctness.
- It is modular, making it easy to update or extend for future changes in tax policies.

This implementation showcases Haskell's capability to handle real-world computational problems in a clean and functional style