

Problem Set # 3

Purpose: Introduction to conditional/selection statements, write program using if-else structure.

Problem 1: People who deal with historical dates use a number called the Julian day to calculate the number of days between two events. The Julian day is the number of days that have elapsed since January 1, 4713 B.C. For example, the Julian day for October 16, 1956, is 2435763.

There are formulas for computing the Julian day from a given date, and vice versa. One very simple formula computes the day of the week from a given Julian day:

Day of the week = (Julian day + 1) % 7 where % is the Java modulus operator. This formula gives a result of 0 for Sunday, 1 for Monday, and so on, up to 6 for Saturday. For Julian day 2435763, the result is 2 (Tuesday). Your job is to write a C++ program that requests and inputs a Julian day, computes the day of the week using the formula, and then displays the name of the day that corresponds to that number.

Your output might look like this:

Enter a Julian day number and press Enter.

2451545

Julian day number 2451545 is a Saturday.

Enter a Julian day number and press Enter.

2451547

Julian day number 2451547 is a Monday.

Problem 2: Write a C++ program of a Body Mass Index (BMI) calculator application that reads the user's weight and height and then calculates and displays the user's body mass index. The formula for calculating BMI is:

English BMI Formula

$$\text{BMI} = (\text{Weight in Pounds} / (\text{Height in inches} \times \text{Height in inches})) \times 703$$

Metric BMI Formula

$$\text{BMI} = (\text{Weight in Kilograms} / (\text{Height in Meters} \times \text{Height in Meters}))$$

Also, the application should display the following information from the scale provided below.

BMI VALUES Underweight: less than 18.5; Normal: between 18.5 and 24.9; Overweight: between 25 and 29.9; Obese: 30 or greater

Note: Your program should check that the height and weight entered are not negative. If the entered data is incorrect display a message accordingly.

Problem 3: A bank in your town updates its customers' accounts at the end of each month. The bank offers two types of accounts: savings and checking. Every customer must maintain a minimum balance. If a customer's balance falls below the minimum balance, there is a service charge of \$10.00 for savings accounts and \$25.00 for checking accounts. If the balance at the end of the month is at least the minimum balance, the account receives interest as follows:

a. Savings accounts receive 4% interest.

b. Checking accounts with balances of up to \$5,000 more than the minimum balance receive 3% interest; otherwise, the interest is 5%.

Problem 4: Write a program that reads a customer's account number (int type), account type (char; s for savings, c for checking), minimum balance that the account should maintain, and current balance. The program should then output the account number, account type, current

balance, and an appropriate message. Test your program by running it five times, using the following data:

```
46728 S 1000 2700
87324 C 1500 7689
79873 S 1000 800
89832 C 2000 3000
98322 C 1000 750
```

Problem 5: Write a C++ program that determines the median of three input numbers. The median is the middle number when the three numbers are arranged in order. However, the user can input the values in any order, so your program must determine which value is between the other two. For example, if the user enters:

41.52; 27.18; 96.03

Then the program would output:

The median of 41.52, 27.18 and 96.03 is 41.52

Once you have the three-number case working, extend the program to handle five numbers.

Note: Be sure to use proper formatting and appropriate comments in your code.

Problem 6: Write a C++ program that asks a user to enter 2 numbers. Assign the numbers to integers 'bigNumber' and 'littleNumber'. Check if bigNumber is bigger than littleNumber. If yes, check if they are evenly divisible, otherwise state, "The littleNumber is bigger."

If they are evenly divisible, display "They are evenly divisible" else display, "They are not evenly divisible". If they are evenly divisible, check if they are the same number. If they are same, display "They are same" else, display "Not same".

Problem 7: Average: Write a program that stores the following values in five separate variables: 40, 32, 17, 29, and 83. The program should calculate the average of the numbers and display it on the screen.

Average = Sum of all the values/ number of values.

Note: Carefully choose the data types to hold data. Which should be int (if any) and which should be float (if any)?

Problem 8: Suppose an employee gets paid every two weeks and earns Rs. 5999.99 each pay period. In a year the employee gets paid 12 times. Write a program that defines the following variables:

1. payAmount: To hold the amount of pay the employee earns each pay period. Initialize the variable with 5999.99
2. payPeriods: To hold the number of pay periods in a year. Initialize the variable with 12.
3. annualPay: To hold the employee's total annual pay, which will be calculated.

The program should calculate the employee's total annual pay by multiplying the employee's pay amount by the number of pay periods in a year, and store the result in the annualPay variable. Display the total annual pay on the screen.

Note: Carefully choose the data types to hold data. Which should be int (if any) and which should be float (if any)?

Problem 9: In a right triangle, the square of the length of one side is equal to the sum of the squares of the length of the other two sides. Write a C++ program that takes in lengths of the three sides of a triangle, and then outputs whether the triangle is right or not.

Problem 10: Write C++ that plays the game of "Rock, paper, scissors." In this game, two players simultaneously say (or display a hand symbol representing) either "rock," "paper," or "scissors." The winner is the one whose choice dominates the other. The rules are: paper dominates (wraps) rock, rock dominates (breaks) scissors, and scissors dominate (cut) paper. Declares and initializes First player and second player variables at the start).

You can use 1=rock, 2=paper,3=scissors

Examples:

First player = 1, second player = 1

Sample Output: Draw

First player = 1, second player = 2

Sample Output: 2nd player wins