SLCC, Application Software Development
ASDV 1220, Programming Fundamentals

MP1, Variables and Arithmetic

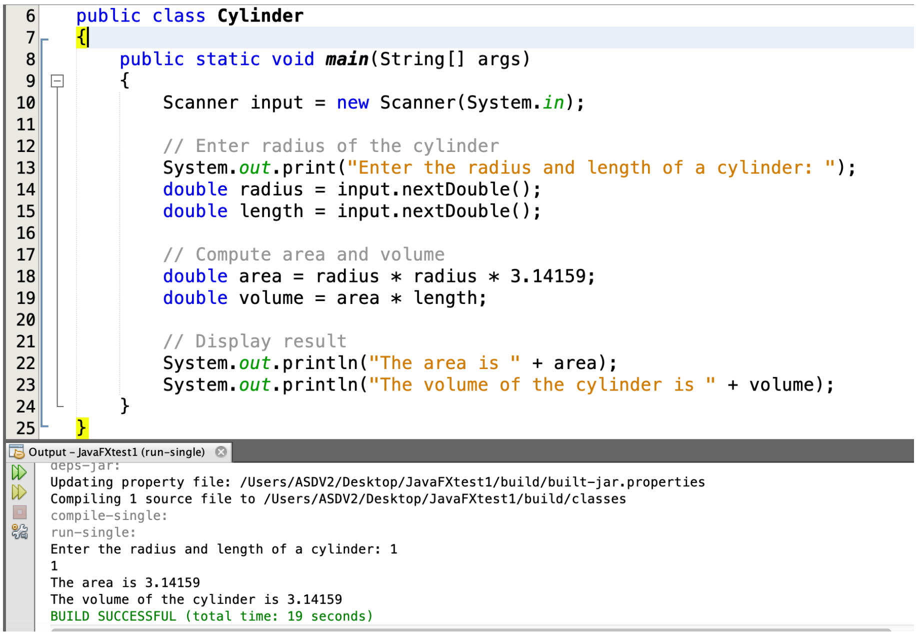
Given solution-problems are worth 20%, the rest 80%

1. **Read the problem carefully, examine carefully the code given and understand it. Implement the class** **Cylinder.**

(Compute the volume of a cylinder)
Write a program that reads in the radius and length of a cylinder and computes the area and volume using the following formulas:

area = radius * radius * π
volume = area * length

The solution at the RHS.

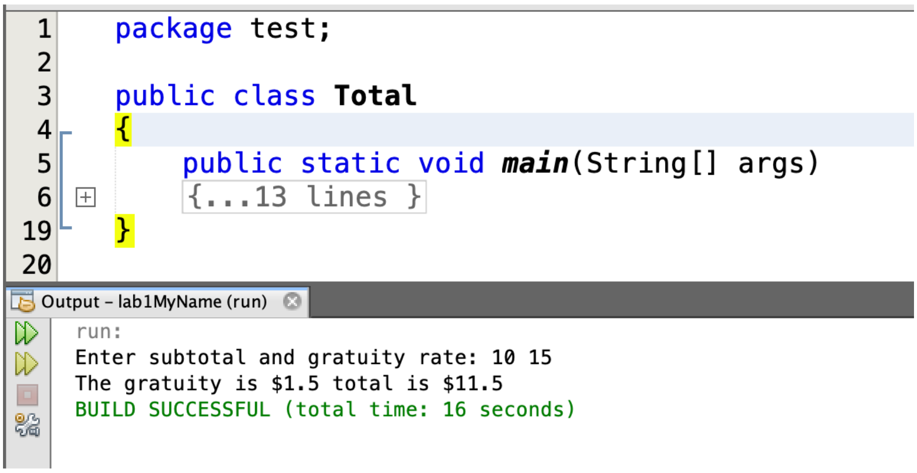


```
public class Cylinder
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        // Enter radius of the cylinder
        System.out.print("Enter the radius and length of a cylinder: ");
        double radius = input.nextDouble();
        double length = input.nextDouble();

        // Compute area and volume
        double area = radius * radius * 3.14159;
        double volume = area * length;

        // Display result
        System.out.println("The area is " + area);
        System.out.println("The volume of the cylinder is " + volume);
    }
}
```

```
deps-jar:
Updating property file: /Users/ASDV2/Desktop/JavaFXtest1/build/built-jar.properties
Compiling 1 source file to /Users/ASDV2/Desktop/JavaFXtest1/build/classes
compile-single:
run-single:
Enter the radius and length of a cylinder: 1
1
The area is 3.14159
The volume of the cylinder is 3.14159
BUILD SUCCESSFUL (total time: 19 seconds)
```

2. Implement the class **Total**

(Financial application: calculate tips) Write a program that reads the subtotal and the gratuity rate, then computes the gratuity and total. For example, if the user enters 10 for subtotal and 15% for gratuity rate, the program displays $1.5 as gratuity and $11.5 as total.



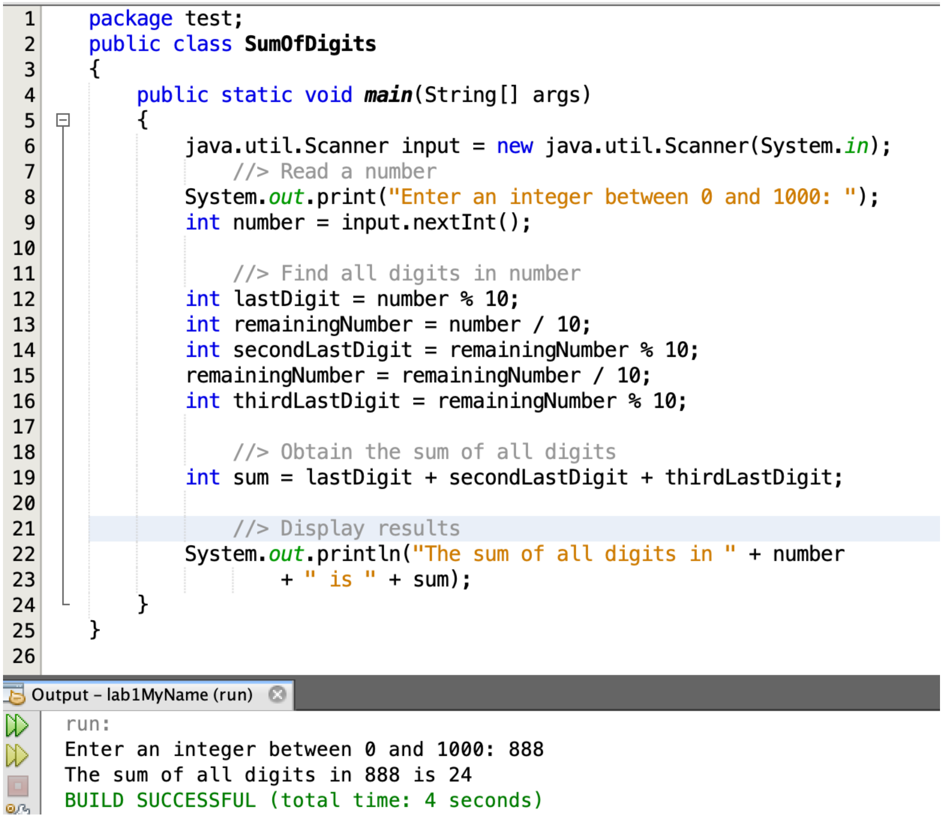
```
package test;

public class Total
{
    public static void main(String[] args)
    {...13 lines }
}

```

```
run:
Enter subtotal and gratuity rate: 10 15
The gratuity is $1.5 total is $11.5
BUILD SUCCESSFUL (total time: 16 seconds)
```

3. **Read the problem carefully, examine carefully the code given and understand it. Implement the class** SumOfDigits

(Sum the digits in an integer) Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14. Hint: Use the % operator to extract digits, and use the / operator to remove the extracted digit. For instance, 932 % 10 = 2 and 932 / 10 = 93.

Solution at RHS.

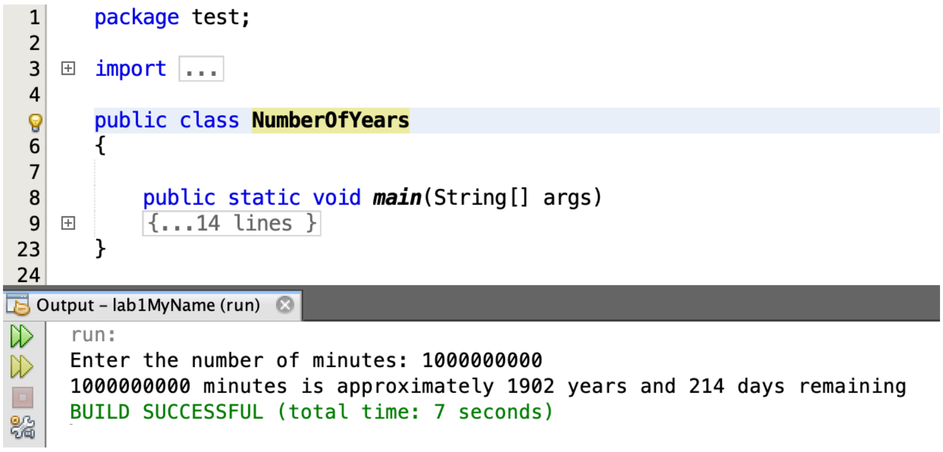


```java
package test;
public class SumOfDigits
{
    public static void main(String[] args)
    {
        java.util.Scanner input = new java.util.Scanner(System.in);
            //> Read a number
        System.out.print("Enter an integer between 0 and 1000: ");
        int number = input.nextInt();

            //> Find all digits in number
        int lastDigit = number % 10;
        int remainingNumber = number / 10;
        int secondLastDigit = remainingNumber % 10;
        remainingNumber = remainingNumber / 10;
        int thirdLastDigit = remainingNumber % 10;

            //> Obtain the sum of all digits
        int sum = lastDigit + secondLastDigit + thirdLastDigit;

            //> Display results
        System.out.println("The sum of all digits in " + number
                + " is " + sum);
    }
}
```

```
run:
Enter an integer between 0 and 1000: 888
The sum of all digits in 888 is 24
BUILD SUCCESSFUL (total time: 4 seconds)
```

4. Implement class **NumberOfYears**

(Find the number of years) Write a program that prompts the user to enter the minutes (e.g., 1 billion), and displays the number of years and days for the minutes. For simplicity, assume a year has 365 days.



```java
package test;

import ...

public class NumberOfYears
{

    public static void main(String[] args)
    {...14 lines }
}
```

```
run:
Enter the number of minutes: 1000000000
1000000000 minutes is approximately 1902 years and 214 days remaining
BUILD SUCCESSFUL (total time: 7 seconds)
```

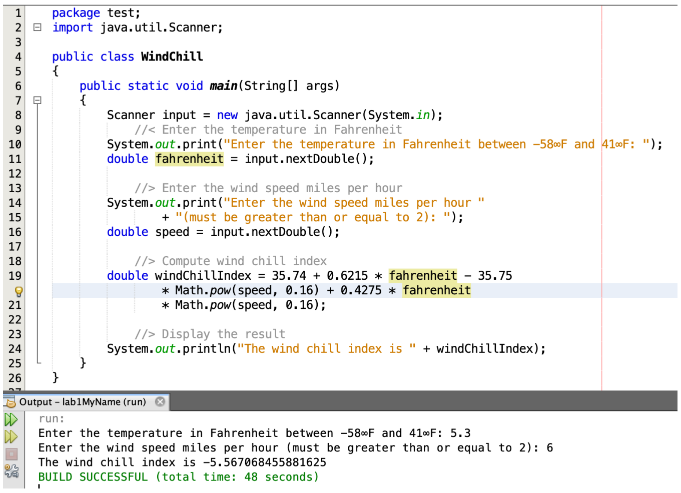
5. **Read the problem carefully, examine carefully the code given and understand it. Implement the class <u>WindChill</u>.**

(Science: wind-chill temperature)
How cold is it outside? The temperature alone is not enough to provide the answer. Other factors including wind speed, relative humidity, and sunshine play important roles in determining coldness outside. In 2001, the National Weather Service (NWS) implemented the new wind-chill temperature to measure the coldness using temperature and wind speed.
The formula is

$$t_{wc} = 35.74 + 0.6215t_a - 35.75v^{0.16} + 0.4275t_a v^{0.16}$$

where t_a is the outside temperature measured in degrees Fahrenheit and v is the speed measured in miles per hour. t_{wc} is the wind-chill temperature. The formula cannot be used for wind speeds below 2 mph or temperatures below -58 degrees F or above 41 degrees F.
Write a program that prompts the user to enter a temperature between -58 degrees F and 41 degrees F and a wind speed greater than or equal to 2 and displays the wind-chill temperature.
Use Math.pow(a, b) to compute $v^{0.16}$.

Solution at RHS.

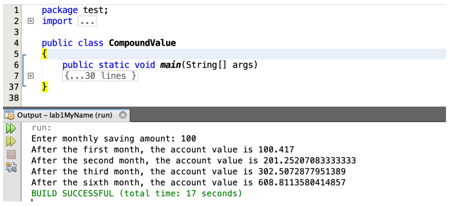


```
package test;
import java.util.Scanner;

public class WindChill
{
    public static void main(String[] args)
    {
        Scanner input = new java.util.Scanner(System.in);
            //< Enter the temperature in Fahrenheit
        System.out.print("Enter the temperature in Fahrenheit between -58∞F and 41∞F: ");
        double fahrenheit = input.nextDouble();

            //> Enter the wind speed miles per hour
        System.out.print("Enter the wind speed miles per hour "
                + "(must be greater than or equal to 2): ");
        double speed = input.nextDouble();

            //> Compute wind chill index
        double windChillIndex = 35.74 + 0.6215 * fahrenheit - 35.75
                * Math.pow(speed, 0.16) + 0.4275 * fahrenheit
                * Math.pow(speed, 0.16);

            //> Display the result
        System.out.println("The wind chill index is " + windChillIndex);
    }
}
```

```
run:
Enter the temperature in Fahrenheit between -58∞F and 41∞F: 5.3
Enter the wind speed miles per hour (must be greater than or equal to 2): 6
The wind chill index is -5.567068455881625
BUILD SUCCESSFUL (total time: 48 seconds)
```

6. **Implement class <u>CompoundValue</u>**

(Financial application: compound value)
Suppose you save $100 each month into a savings account with the annual interest rate 5%. Thus, the monthly interest rate is 0.05/12 = 0.00417. After the first month, the value in the account becomes
100 * (1 + 0.00417) = 100.417
After the second month, the value in the account becomes
(100 + 100.417) * (1 + 0.00417) = 201.252
After the third month, the value in the account becomes
(100 + 201.252) * (1 + 0.00417) = 302.507
and so on.
Write a program that prompts the user to enter a monthly saving amount and displays the account value after the 1st, 2nd....6th month.



```
package test;
import ...

public class CompoundValue
{
    public static void main(String[] args)
    {...30 lines }
}
```

```
run:
Enter monthly saving amount: 100
After the first month, the account value is 100.417
After the second month, the account value is 201.25207083333333
After the third month, the account value is 302.5072877951389
After the sixth month, the account value is 608.8113580414857
BUILD SUCCESSFUL (total time: 17 seconds)
```