COMSATS University, Islamabad (CUI)

Department of Computer Science

Assignment-1, Fall 2024

Solution

[CLO-1]: Demonstrate the fundamental concepts of programming

Course: CSC103 – Programming Fundamentals **Class: BCS-2B**
Submission Deadline: October 15, 2024 **Total Marks: 10**

Upon completion of this assignment, students will be able to understand:

- **how to choose, declare, and use different Java data types**
- **how to write and evaluate expressions in Java using various operators**

Mastering these concepts is foundational for writing more complex and functional Java programs

For this assignment, the submission should include:

- **A well- formatted *Word* file containing solutions for Questions 1 and 2.**
- **Physics.java file for Question 3**
- **Ohm.java file for Question 4**

Question 1: Data Types and Binary Encoding

Part a)- Assign all valid literals to each of the following Java data types (One to many mappings possible from *Data type* to *Literal*). To do this, think of one variable from the specified data type, and then consider whether you can assign this variable to the literal. Finally, write a Java statement that would declare and initialize the variable with the corresponding literal. **[1.5]**

Data type
`byte`
`short`
`int`
`long`
`float`
`double`
`char`
`boolean`
`String`

Literal
`-9_223_372_036_854_775_808L`
`True`
`-1_2_8`
`"true"`
`+32768`
`'\uffff'`
`-987_654.321e-2_3f`
`-0B1111111_11111111`
`-.1e+111`
`'\"'`

Solution:

```
long long_var = -9_223_372_036_854_775_808L;

 boolean boolean_var = true;

 byte byte_var = -1_2_8;

 String string_var = "true";
```

```
int int_var = +32768;

char char_var = '\uffff';

float float_var = -987_654.321e-2_3f;

short short_var = -0B1111111_11111111;

double double_var = -.1e+111;

char another_char_var = '\"';
```

Part b)- For each of the following sets, determine the number of bits required to identify each element of the set with a unique bit sequence and name the smallest (just suitable) Java data type: **[1.5]**

- Seats in the Rawalpindi Cricket Stadium
- Number of students in our class
- People on Earth at the beginning of 2024
- Seconds in a week
- Position of a light switch (no dimmer)
- Theoretically possible car plates for motor vehicles (Car Plate Format: Three letters in combination with four digits)

Solution:

- Seats in the Rawalpindi Cricket Stadium

Rawalpindi Cricket Stadium has a capacity of around 15000. Since $\log_2 15000 \approx 13.9$, so minimum number of bits required to represent 15000 as a signed number are 14+1 = 15. The suitable Java data type is short.

- Number of students in our class

Since $\log_2 50 \approx 5.6$, so minimum number of bits required to represent 50 as a signed number are 6+1 = 7. The suitable Java data type is byte.

- People on Earth at the beginning of 2024

A quick google search says that the total number of people is around 8.2 billion. Since $\log_2 8200000000 \approx 32.93$, so minimum number of bits required to represent 8200000000 as a signed number are 33+1 = 34. The suitable Java data type is long.

- Seconds in a week

There are 604800 seconds in a week. Since $\log_2 604800 \approx 19.2$, so minimum number of bits required to represent 604800 as a signed number are 20+1 = 21. The suitable Java data type is int.

- Position of a light switch

Since the switch can either be on or off, we can represent it using only one bit. It could be 0 for off and 1 for on. The suitable Java data type is Boolean.

- Theoretically possible car plates for motor vehicles (Car Plate Format: Three letters in combination with four digits)

Assuming that last four digits cannot be zero altogether, the total possibilities are = 26 x 26 x 26 x 10 x 10 x 10 x 10 -1 = 175759999. Since $\log_2 175759999 \approx 27.4$, so minimum number of bits required to represent 175759999 as a signed number are 28+1 = 29. The suitable Java data type is int.

Question 2: Data Types and Expressions

Given the following code section:

```
int x = 23, y = 12;
double z = 0.815;
```

Evaluate each of the following expressions individually, and immediately after the above code, by finding the value of the right-hand side. In addition, specify the data type each variable must have, to which the value of the expression is assigned**. [1.5]**

a)- `a = (y > 15) && (5 != 4) || (x < y);`

b)- `b = +x++ + ++x - -x-- - - -x;`

c)- `c = -0.815;`
`c *= z;`

d)- `d = x | y << 20;`

e)- `e = ((x == y) || (x < z * y));`

f)- `f = x << x;`

g)- `g = !((x >> 666 < y) & (z++ == x));`

h)- `h = 4711 << x;`

Solution:

a) `a = (y > 15) && (5 != 4) || (x < y);`
Right hand side evaluates to `false`. The data type of variable a should be `boolean`
b) `b = +x++ + ++x - -x-- - --x;`
Right hand side evaluates to `50`. The data type of variable b should be `int`
c) `c = -0.815;`
`c *= z;`
Right hand side evaluates to `-0.664225`. The data type of variable c should be `double`
d) `d = x | y << 20;`
Right hand side evaluates to `12582935`. The data type of variable d should be `int`
e) `e = ((x == y) || (x < z * y));`
Right hand side evaluates to `false`. The data type of variable e should be `boolean`
f) `f = x << x;`
Right hand side evaluates to `192937984`. The data type of variable f should be `int`
g) `g = !((x >> 666 < y) & (z++ == x));`
Right hand side evaluates to `true`. The data type of variable g should be `boolean`
h) `h = 471l << x;`
Right hand side evaluates to `3951034368`. The data type of variable h should be `long`

Question 3: Physics - Thermal Equilibrium [3]

For ideal gases the following thermal equilibrium is valid: (Pressure p, Volume V, Particle number N, Boltzmann-Constant k_B, Temperature T)

$$p \,.\, V = N \,.\, k_B \,.\, T$$

The substance quantity N is always assumed to be 1 mol for this question, so $N = N_A$, where N_A is the so called Avogadro-Constant. Furthermore, the average kinetic energy of a gas particle (mass m, velocity v) is calculated as:

$$E_{kin} = ½ \,.\, m \,.\, v^2 = \frac{3}{2} \,.\, k_B \,.\, T$$

In this question, you are going to write expressions in Java using above formulas. You have been provided with three files:

- *Physics.java*: This is the only file in which you need to add the code comprising expressions. The expressions are to be added in the methods at the corresponding lines marked with comments labeled "TODO".
- *PhysicsConstants.java*: This file contains constants that would be needed in the expressions.
- *PhysicsPublicTest.java*: This file contains test cases to test the code you will be writing in Physics.java. This will really help you in getting immediate feedback on the correctness of your code.

Solution:

```
public class Physics {
       // Given gas-volume v (m*m*m) and temperature t (K),
       // this method computes and returns the pressure p (Pa).
       public static double computeP(double v, double t) {
              //p = (NA. kB. t)/v
               return (PhysicsConstants.AVOGADRO*PhysicsConstants.BOLTZMANN*t)/v;
       }

       // Given pressure p (Pa) and temperature t (K),
       // this method computes and returns the gas-volume v (m*m*m).
       public static double computeV(double p, double t) {
              //v = (NA. kB. t)/p
               return (PhysicsConstants.AVOGADRO*PhysicsConstants.BOLTZMANN*t)/p;
       }

       // Given pressure p (Pa) and gas-volume v (m*m*m),
       // this method computes and returns the temperature t (K).
       public static double computeT(double p, double v) {
              //t = (p.v)/(NA. kB)
               return (p*v)/(PhysicsConstants.AVOGADRO*PhysicsConstants.BOLTZMANN);
       }

       // Given gas-volume v (m*m*m) and a change in temperature of deltaT (K),
       // this method computes and returns the change in pressure (Pa).
       public static double computeDeltaPisochore(double v, double deltaT) {
              // deltaP = (NA. kB. deltaT)/v
               return (PhysicsConstants.AVOGADRO*PhysicsConstants.BOLTZMANN*deltaT)/v;
       }

       // Given gas-volume v (m*m*m), temperature t (K),and a change in volume of deltaV
       // (m*m*m), this method computes and returns the change in pressure (Pa).
       public static double computeDeltaPisotherm(double v, double t, double deltaV) {
```

```
                //deltaP = -(NA. kB. t)/(2*deltaV)
                return -(PhysicsConstants.AVOGADRO*PhysicsConstants.BOLTZMANN*t)/(2*deltaV);
        }

        // Given temperature t (K) and molar mass of particle m (kg/mol),
        // this method computes and returns the average speed of a particle.
        public static double computeAverageSpeed(double t, double m) {
                // 1/2 *m * v*v = 3/2 . kB . T
                return Math.sqrt((3*PhysicsConstants.BOLTZMANN*t*PhysicsConstants.AVOGADRO)/m);
        }
}
```

Question 4: Exciting Resistors [2.5]

In a closed circuit, Ohm's law applies: (voltage V, resistance R, current I)

$$V = R \cdot I$$

For a series connection of 2 resistors, the total resistance is calculated as:

$$R_s = R_1 + R_2$$

For a parallel connection of 2 resistors, the total resistance is calculated as:

$$R_p = \frac{R1 \cdot R2}{R1 + R2}$$

In this question, you are going to write expressions in Java using above formulas. You have been provided with two files:

- *Ohm.java*: This is the only file in which you need to add the code comprising expressions. The expressions are to be added in the methods at the corresponding lines marked with comments labeled "TODO".
- *OhmPublicTest.java*: This file contains test cases to test the code you will be writing in Ohm.java. This will really help you in getting immediate feedback on the correctness of your code.

Solution:

```
/* Ohm's law: U = R*I */
public class Ohm {
        /* Calculate the voltage using Ohm's law */
        public static double voltage(double resistance, double current) {
             // TODO
             return resistance*current;
        }

        /* Calculate the current using Ohm's law */
        public static double current(double voltage, double resistance) {
             // TODO
             return voltage/resistance;
        }

        /* Calculate the resistance using Ohm's law */
        public static double resistance(double voltage, double current) {
             // TODO
             return voltage/current;
        }
        /* Calculate the resistance of two serial connected resistances */
        public static double resistanceSerialConnection2(double r1, double r2) {
             // TODO
             return r1+r2;
        }

        /* Calculate the resistance of two parallel connected resistances */
        public static double resistanceParallelConnection2(double r1, double r2) {
             // TODO
             return (r1*r2)/(r1+r2);
        }
}
```