PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

Course Code: CCE-121

SUBMITTED TO:

Dr. Md Samsuzzaman Sobuz

Department of Computer And Communication Engineering

Faculty of Computer Science And Engineering

SUBMITTED BY:

Name: Md Noushad Bhuiyan

ID: 2102038 , Registration No: 10165

Faculty of Computer Science and Engineering

Date of submission: 13/11/23

Chapter 06

6.1 Fill in the blanks in each of the following statements:

a. Method Call

b. Local Variable

c. Return

d. void

e. Top

f. Last in First out(LIFO)

g. return; or return expression; or encountering the closing right brace of a method;

h. SecureRandom

i. StackFrame or Activation Record

j. StackOverflow

k. Scope

l. Method Overloading

6.2 For the class Craps in Fig. 6.8, state the scope of each of the following entities:

a) the variable randomNumbers.

b) the variable die1.

c) the method rollDice.

d) the method main.

e) the variable sumOfDice

a. class body.

b. block that defines method rollDice's body.

c. class body.

d. class body.

e. block that defines method main's body

6. 3 Write an application that tests whether the examples of the Math class method calls shown in Fig. 6.2 actually produce the indicated results.

```
public class MathTest
{
 public static void main(String[] args)  {
System.out.printf("Math.abs(23.7) = %f%n", Math.abs(23.7));
System.out.printf("Math.abs(0.0) = %f%n", Math.abs(0.0));
 System.out.printf("Math.abs(-23.7) = %f%n", Math.abs(-23.7));
 System.out.printf("Math.ceil(9.2) = %f%n", Math.ceil(9.2));
 System.out.printf("Math.ceil(-9.8) = %f%n", Math.ceil(-9.8));
 System.out.printf("Math.cos(0.0) = %f%n", Math.cos(0.0));
 System.out.printf("Math.exp(1.0) = %f%n", Math.exp(1.0));
 System.out.printf("Math.exp(2.0) = %f%n", Math.exp(2.0));
 System.out.printf("Math.floor(9.2) = %f%n", Math.floor(9.2));
 System.out.printf("Math.floor(-9.8) = %f%n", Math.floor(-9.8));
 System.out.printf("Math.log(Math.E) = %f%n", Math.log(Math.E));
System.out.printf("Math.log(Math.E * Math.E) = %f%n",
 Math.log(Math.E * Math.E));
 System.out.printf("Math.max(2.3, 12.7) = %f%n", Math.max(2.3, 12.7));
System.out.printf("Math.max(-2.3, -12.7) = %f%n",
 Math.max(-2.3, -12.7));
 System.out.printf("Math.min(2.3, 12.7) = %f%n", Math.min(2.3, 12.7));
 System.out.printf("Math.min(-2.3, -12.7) = %f%n",
 Math.min(-2.3, -12.7));
System.out.printf("Math.pow(2.0, 7.0) = %f%n", Math.pow(2.0, 7.0));
System.out.printf("Math.pow(9.0, 0.5) = %f%n", Math.pow(9.0, 0.5));
System.out.printf("Math.sin(0.0) = %f%n", Math.sin(0.0));
 System.out.printf("Math.sqrt(900.0) = %f%n", Math.sqrt(900.0));
 System.out.printf("Math.tan(0.0) = %f%n", Math.tan(0.0));  } }
```

Output:

```
Math.abs(23.7) = 23.700000

Math.abs(0.0) = 0.000000

Math.abs(-23.7) = 23.700000

Math.ceil(9.2) = 10.000000

Math.ceil(-9.8) = -9.000000

Math.cos(0.0) = 1.000000

Math.exp(1.0) = 2.718282

Math.exp(2.0) = 7.389056

Math.floor(9.2) = 9.000000

Math.floor(-9.8) = -10.000000

Math.log(Math.E) = 1.000000

Math.log(Math.E * Math.E) = 2.000000

Math.max(2.3, 12.7) = 12.700000

Math.max(-2.3, -12.7) = -2.300000

Math.min(2.3, 12.7) = 2.300000

Math.min(-2.3, -12.7) = -12.700000

Math.pow(2.0, 7.0) = 128.000000

Math.pow(9.0, 0.5) = 3.000000

Math.sin(0.0) = 0.000000

Math.sqrt(900.0) = 30.000000

Math.tan(0.0) = 0.000000
```

6.4 Give the method header for each of the following methods:

a) Method hypotenuse, which takes two double-precision, floating-point arguments side1 and side2 and returns a double-precision, floating-point result.

b) Method smallest, which takes three integers x, y and z and returns an integer.

c) Method instructions, which does not take any arguments and does not return a value. [Note: Such methods are commonly used to display instructions to a user.]

d) Method intToFloat, which takes integer argument number and returns a float

Answer:

a) double hypotenuse(double side1, double side2)

b) int smallest(int x, int y, int z)

c) void instructions()

d) float intToFloat(int number)

6.5 Find the error in each of the following program segments. Explain how to correct the error.

a) Error: Method h is declared within method g. Correction: Move the declaration of h outside the declaration of g.

b) Error: The method is supposed to return an integer, but does not. Correction: Delete the variable result, and place the statement return x + y; in the method, or add the following statement at the end of the method body: return result;

c) Error: The semicolon after the right parenthesis of the parameter list is incorrect, and the parameter a should not be redeclared in the method. Correction: Delete the semicolon after the right parenthesis of the parameter list, and delete the declaration float a;.

d) Error: The method returns a value when it's not supposed to. Correction: Change the return type from void to int.

6.6 Declare method sphereVolume to calculate and return the volume of the sphere. Use the following statement to calculate the volume: double volume = (4.0 / 3.0) * Math.PI * Math.pow(radius, 3)

```
// Calculate the volume of a sphere.

import java.util.Scanner;

public class Sphere

{

// obtain radius from user and display volume of sphere

public static void main(String[] args)  {

Scanner input = new Scanner(System.in);

System.out.print("Enter radius of sphere: ");

double radius = input.nextDouble();
System.out.printf("Volume is %f%n", sphereVolume(radius));  } // end method determineSphereVolume
// calculate and return sphere volume
public static double sphereVolume(double radius)  {
double volume = (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
return volume;  } // end method sphereVolume
} // end class Sphere
```

6.7 Answer:

a. 7.5

b. 7

c. 12

d. -5

e. 9

f. -11

g. -6

6.8 (Parking Charges) A parking garage charges a $2.00 minimum fee to park for up to three hours. The garage charges an additional $0.50 per hour for each hour or part thereof in excess of three hours. The maximum charge for any given 24-hour period is $10.00. Assume that no car parks for longer than 24 hours at a time. Write an application that calculates and displays the parking charges for each customer who parked in the garage yesterday. You should enter the hours parked for each customer. The program should display the charge for the current customer and should calculate and display the running total of yesterday’s receipts. It should use the method calculateCharges to de□termine the charge for each customer.

```
import java.util.Scanner;

public class Garage_parking {
    public static void main(String[] args) {
        Scanner s= new Scanner(System.in);
        System.out.println("Enter the parking hours:");
        double hour=s.nextDouble();
        garageBill(hour);

    }
    static void garageBill(double hour)
{
    double value;
        if(hour<=3)
        {
            System.out.println("Total cost is:  2 Dollars");
        }
        else if(hour>3 && hour<=24)
        {
            value=2+((hour-3)*0.5);
            if(value>10)
            System.out.println("Total cost is: 10 Dollars");
            else
            System.out.println("Total cost is: "+value+" Dollars");
        }
        else if(hour>24)
        System.out.println("You cannot park a vehicle more than 24 hours");
}
}
```

6.9 (Rounding Numbers)

```
import java.util.Scanner;

class Solution
{
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
double x = scanner.nextDouble();
scanner.close();

double y = Math.floor(x + 0.5);
System.out.println("Original: " + x + ", Rounded: " + y);
}
}
```

6.10 (Rounding Numbers)

```
import java.util.Scanner;


public class Solution
{
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
double x = scanner.nextDouble();
scanner.close();

System.out.println("Original: " + x + "\n" +
"Round to Integer: " + roundToInteger(x) + "\n" +
"Round to Tenths: " + roundToTenths(x) + "\n" +
"Round to Hundredths: " + roundToHundredths(x) + "\n" +
"Round to Thousandths: " + roundToThousandths(x));
}

public static double roundToInteger(double x) {
return Math.floor(x + 0.5);
}

public static double roundToTenths(double x) {
return Math.floor(x * 10 + 0.5) / 10; }

public static double roundToHundredths(double x) {
return Math.floor(x * 100 + 0.5) / 100;
}

public static double roundToThousandths(double x) {
return Math.floor(x * 1000 + 0.5) / 1000;
}
}
```

6.11 Answer each of the following questions:

a) What does it mean to choose numbers “at random”?

Ans: Choosing at random means to choose a integer or floating point number without learning it's value randomly.

b) Why is the nextInt method of class SecureRandom useful for simulating games of chance?

Ans: Random can be useful to develop neutral mobs and mods in games, like in a car racing game, the direction of other bot cars.

c) Why is it often necessary to scale or shift the values produced by a SecureRandom object?

Ans: Scalling or shifting may be necessary to get the output in between a number or a range.

d) Why is computerized simulation of real-world situations a useful technique?

Ans: With simulations we can observe real world situations and other complex systems, which enables researchers to learn more about a system.

6.12 Write statements that assign random integers to the variable n in the following ranges:

a) 2 ≤ n≤ 6

```
> int n = random.nextInt(5) + 2;
```

b) 4≤ n ≤ 50

```
> int n = random.nextInt(47) + 4;
```

c) 0≤ n≤ 7

```
> int n = random.nextInt(8);
```

d) 1000 ≤ n ≤ 1030

```
> int n = random.nextInt(31) + 1000;
```

e) –5≤ n≤ 1

```
> int n = random.nextInt(7) - 5;
```

f) –2 ≤ n ≤ 9

```
> int n = random.nextInt(12) - 2;
```

6.13 Write statements that will display a random number from each of the following sets:

a) 0, 3, 6, 9, 12.

```
n = (int) (Math.random() * 5) * 3;
```

b) 1, 2, 4, 8, 16, 32.

```
n = (int) Math.pow(2, (int) (Math.random() * 6));
```

c) 10, 20, 30, 40.

```
n = (int) (Math.random() * 4 + 1) * 10;
```

6.14 (Floor and Ceil)

```
public class Round {
public static void main(String[] args) {
```

```
Round round = new Round();

System.out.println(round.myFloor(3.5));
System.out.println(round.myFloor(-3.5));
System.out.println(round.myCeil(-3.5));
System.out.println(round.myCeil(-3.6));
}

double myFloor(double x) {
return Math.floor(x);
}

double myCeil(double x) {
return Math.ceil(x);
}
}
```

6.15 (Hypotenuse Calculations)

Ans:

```
public class Triangle {
public static void main(String[] args) {
double side1, side2, hypotenuse;
side1 = 3.0;
side2 = 4.0;
hypotenuse = hypotenuse(side1, side2);
System.out.println("Given sides of lengths " + side1 + " and " + side2 + " the hypotenuse is " + hypotenuse);
side1 = 5.0; side2 = 12.0;
hypotenuse = hypotenuse(side1, side2);
System.out.println("Given sides of lengths " + side1 + " and " + side2 + " the hypotenuse is " + hypotenuse);

side1 = 8.0;
side2 = 15.0;
hypotenuse = hypotenuse(side1, side2);
System.out.println("Given sides of lengths " + side1 + " and " + side2 + " the hypotenuse is " + hypotenuse);
}
static double hypotenuse(double a, double b) {
return Math.sqrt(a * a + b * b);
}
}
```