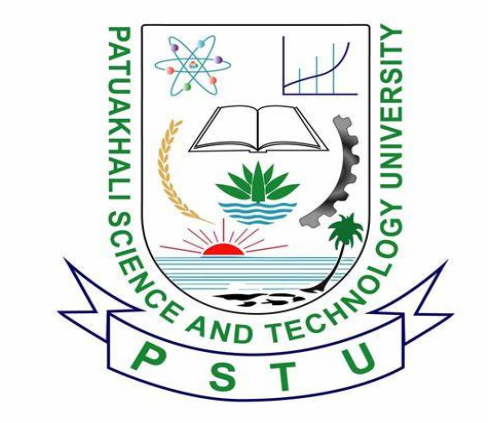
Patuakhali Science and Technology University



Course Code: CCE-112

SUBMITTED TO:

Dr. Md Samsuzzaman Sobuz

Department of Computer And Communication Engineering

Faculty of Computer Science And Engineering

SUBMITTED BY:

Name: Tanmoy Kumar Das

ID: 2102079, Registration No: 10206

Faculty of Computer Science and Engineering

Date of submission: 31/10/2023

5.1)Fill in the blanks in each of the following statements:

a) Typically, for statements are used for counter-controlled repetition and while statements for sentinel-controlled repetition.

b) The do...while statement tests the loop-continuation condition after executing the loop's body; therefore, the body always executes at least once.

c) The switch statement selects among multiple actions based on the possible values of an integer variable or expression, or a String.

d) The continue statement, when executed in a repetition statement, skips the remaining statements in the loop body and proceeds with the next iteration of the loop.

e) The and operator can be used to ensure that two conditions are both true before choosing a certain path of execution.

f) If the loop-continuation condition in a for header is initially false, the program does not execute the for statement's body.

g) Methods that perform common tasks and do not require objects are called static methods.

5.2 State whether each of the following is true or false. If false, explain why:

a) The default case is required in the switch selection statement.

Ans: False. It's optinal.

b) The break statement is required in the last case of a switch selection statement.

Ans: False. Without break the rest of the statements work but it's optional.

c) The expression ((x > y) && (a < b)) is true if either x > y is true or a < b is true.

Ans: False. For conditional and operator both of the operands have to be true.

d) An expression containing the || operator is true if either or both of its operands are true.

Ans: True.

e) The comma (,) formatting flag in a format specifier (e.g., %,20.2f) indicates that a value should be output with a thousands separator.

Ans: True.

f) To test for a range of values in a switch statement, use a hyphen (–) between the start and end values of the range in a case label.

Ans: False. We cannot test a range of values in switch-case statement.

g) Listing cases consecutively with no statements between them enables the cases to perform the same set of statements.

Ans: True.

5.3 Write a Java statement or a set of Java statements to accomplish each of the following tasks:

a) Sum the odd integers between 1 and 99, using a for statement. Assume that the integer variables sum and count have been declared.

```
1 sum = 0;
2 for (count = 1; count <= 99; count += 2)
3 sum += count;
```

b) Calculate the value of 2.5 raised to the power of 3, using the pow method.

```
1 double result = Math.pow(2.5, 3);
```

c) Print the integers from 1 to 20, using a while loop and the counter variable

```
 1 i = 1;
 2 while (i <= 20)
 3 {
 4 System.out.print(i);
 5 if (i % 5 == 0)
 6 System.out.println();
 7 else
 8 System.out.print('\t');
 9 ++i;
10 }
```

d) Repeat part (c), using a for statement.

```
1 for (i = 1; i <= 20; i++)
2 {
3 System.out.print(i);
4 if (i % 5 == 0)
5 System.out.println();
6 else
7 System.out.print('\t');
8 }
```

5.4 Find the error in each of the following code segments, and explain how to correct it:

```
1 a) i = 1;
2 while (i <= 10);
3 ++i;
4 }
```

Ans: There's a semocolon after while header which will cause an infinity loop and a left brace is missing.

```
1 b) for (k = 0.1; k != 1.0; k += 0.1)
2 System.out.println(k);
```

Ans: Using for with a floating-point may not work, because it's precision is different in different machines.

c)

```
 1 switch (n)
 2 {
 3 case 1:
 4 System.out.println("The number is 1");
 5 case 2:
 6 System.out.println("The number is 2");
 7 break;
 8 default:
 9 System.out.println("The number is not 1 or 2");
10 break;
11 }
```

Ans: First case is missing a break block. The code will also work but it's recommended to avoid logical errors.

d) The following code should print the values 1 to 10:

```
1 n = 1;
2 while (n < 10)
3 System.out.println(n++);
```

Ans: The loop will work to print the values between 1 to 9 and will not print 9 becuase of < operator. So we can use while (n < 11) to avoid this error.

5.5 Describe the four basic elements of counter-controlled repetition.

Ans: The four basic elements are, a control variable, the initial value, the increment and the loop-continuation condition.

A Control variable: It is a variable and the loop depends on this variable.

The initial value: It refers to the initial value of the control variable.

The increment: It refers to the change that happens on control variable on each loop.

The loop-continuation condition: It's the condition of the loop.

5.6) Compare and contrast the while and for repetition statements.

The main difference between for repetition and while loop is how the basic four elements of counter-controlled repetition is applied. In while loop, we can only define the loop-continuation condition, where in the for loop we can define control variable, initialize the value, loop-continuation condition as well as the increment.

5.7) If you need to execute the body of a loop at least once, would it be better to use a do...while statement or a while statement?

Ans: For executing the body of the loop at least once, most better option will be do..while statement over while statement. Because in the do..while statement, the inner body will be executed before checking the loopcontinuation condition but in the while loop that will occur in the beginning.

5.8)Compare and contrast the break and continue statements.

Ans: The main contrast between these two are that break statement will exit the loop from the occurance and the rest of the lines will be ignored as well as loop-continuation statement. On the other hand, continue will re-run the loop.It'll also avoid the rest of the lines will be ignored as well as loop-continuationstatement, but in this time the loop will run again.

5.9)Find and correct the error(s) in each of the following segments of code:

1 a) while (i = 1; i <= 10, i+)

2 System.out.println(i);

Ans: Here, the increment part has a logical error. It should be i++.

b) The following code should print whether an integer value is negative or zero:

1 switch (value)

2 {

3 Case value < 0:

4 System.out.println("Negative");

```
5 case 0:
6 System.out.println("Zero");
7 }
```

Ans: We can't use control operator in switch-case.

c) The following code should output the odd integers from 19 to 1:

```
1 for (int i = 19; i > 1; i =+ 1)
2 System.out.println(i);
```

Ans: Here the increment part should be i -= 1 and the loop-continuation should be i >= 1;

d) The following code should output the even integers from 1 to 50:

```
1 counter = 0;
2 do
3 {
4 System.out.println(counter + 1);
5 counter += 2;
6 } while (counter <= 51);
```

Ans: Here the first initialize point of counter should be counter = 1. And the condition should be (counter <= 49)

5.10 What does the following program do?

```
 1 // Exercise 5.10: Printing.java
 2 public class Counting {
 3 public static void main( String[] args )
 4 {
 5 Scanner s = new Scanner(System.in);
 6 for (int i = 1; i < 3; i++)
 7 {
 8 for (int j = 1; j < 5; j++)
 9 System.out.print('*');
10 System.out.println("\n#####");
11 } // end outer for loop
12 } // end main
13 } // end class Counting
```

Ans: Here the code will print,

****

#####

****

#####

5.11 (Extremes) Write an application that finds the minimum and maximumamongst several integers and then computes the sum of the two extremes.The user will be prompted to input how many values the application shouldask the user to input.

Ans:

```
 1 import java.util.Scanner;
 2
 3 public class Main {
 4 public static void main(String[] args) {
 5 Scanner sc = new Scanner(System.in);
 6 System.out.print("How many integers you want to check? ");
 7 int n = sc.nextInt();
 8
 9 int arr[] = new int[n];
10 for (int i=0; i <n; i++)
11 arr[i] = sc.nextInt();
12
13 int max = arr[0], min = arr[0];
14 for (int i=0; i <n; i++)
15 {
16 if (arr[i] > max)
17 max = arr[i];
18 if (arr[i] < min)
19 min = arr[i];
20 }
21 sc.close();
22
23 System.out.println("Max : " + max);
24 System.out.println("Min : " + min);
```

```
25 System.out.println("Sum : " + (max + min));
26 }
27 }
```

5.12 (Integers Divisible by 3) Write an application that calculates the sum of those integers between 1 and 30 that are divisible by 3.

Ans:

```
1 public class Main {
2 public static void main(String[] args) {
3 var sum = 0;
4 for (int i = 1; i <= 30; i++)
5 if (i % 3 == 0)
6 sum += i;
7 System.out.println(sum);
8 }
9 }
```

5.13 (The Sum of a Series) Find the summation of the sequence of numbers1, 2, 3 ... n, where n ranges from 1 to 100. Use type long. Display the results in a tabular format that shows n and the corresponding sum. If this were a product instead of a sum, what difficulty might you encounter with the variable that accumulates the product?

Ans:

```
 1 public class Main {
 2 public static void main(String[] args) {
 3 var sum = 0;
 4 for (int i = 1; i <= 100; i++) {
 5 System.out.print(i + " + ");
 6 sum += i;
 7 }
 8 System.out.println(" = " + sum);
 9 }
10 }
```

5.14 (Modified Compound-Interest Program)

Ans:

```
1 .public class Interest {
```

```
2 .public static void main(String[] args) {
3 .double amount; // amount on deposit at end of each year
4 .double principal = 1000.0; // initial amount before interest
5 .double rate = 5; // interest rate
6 .// display headers
7. System.out.printf("%s%20s%n", "Year", "Amount on deposit");
8 .
9. for (; rate < 11; rate++) {// calculate amount on deposit for each of ten years
10 .System.out.println("Rate = " + rate);
11. for (int year = 1; year <= 10; ++year) {
12 .// calculate new amount for specified year
13. amount = principal * Math.pow(1.0 + rate, year);
14 .// display the year and the amount
15. System.out.printf("%4d%,20.2f%n", year, amount);
16. }
17. }
18 .}
19. }
```

5.15 (Triangle Printing Program)

Ans:

```
1 public class Main {
2 public static void main(String[] args) {
3 int n = 10;
4
5 for (int i = 1; i <= n; i++) {
6 for (int j = 1; j <= i; j++) {
7 System.out.print("*");
8 }
9 System.out.println();
10 }
11
```

```
12 System.out.println();
13
14 for (int i = n; i >= 1; i--) {
15 for (int j = 1; j <= i; j++) {
16 System.out.print("*");
17 }
18 System.out.println();
19 }
20
21 System.out.println();
22
23 for (int i = n; i >= 1; i--) {
24 for (int j = n; j >= i; j--) {
25 System.out.print(" ");
26 }
27 for (int j = 1; j <= i; j++) {
28 System.out.print("*");
29 }
30 System.out.println();
31 }
32
33 System.out.println();
34
35 for (int i = 1; i <= n; i++) {
36 for (int j = n; j >= i; j--) {
37 System.out.print(" ");
38 }
39 for (int j = 1; j <= i; j++) {
40 System.out.print("*");
41 }
42 System.out.println();
```

```
43 }
44 }
45 }
```