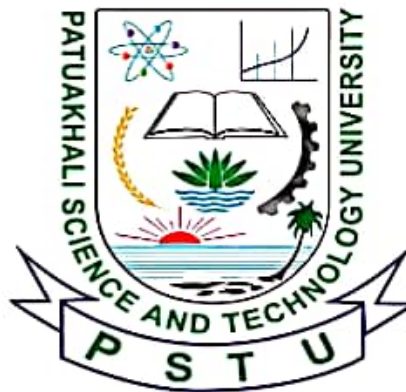


PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY



SUBMITTED TO:

Md. Mahbubur Rahman

Department of Computer Science And Information
Technology

Faculty of Computer Science And Engineering

SUBMITTED BY:

Prosenjit Mondol

I'd no : 2102049 ; Reg no : 10176

Assignment no: 03, Chapter 6 Solution (Theory)

Submitted date: 26-03-2023

Multiple Choice Questions

6.2 Nesting is allowed for which of the following statements? Ans: All of the above.

6.3 What will be the output of the following if-else statement?

Ans: x is assigned the value 5 and string 'Condition is true' is printed.

6.4 For the following if-else construct.

Ans: True if $5 \leq a \leq 10$, false otherwise.

6.5 For the following if-else construct,
if ($x \% 3 == 0$ & $x \% 7 == 0$)

Ans: True if the given positive integer x is a multiple of 3 and 7; false otherwise.

6.6 What will be the output of the following program?

Ans: String 'Hello World' is printed.

2

Review Questions

6.1 True or false:

- (a) A switch expression can be of any type. (false)
- (b) A program stops its execution when a break statement is encountered. (false)
- (c) Each case label can have only one statement. (True)
- (d) The default case is required in the switch statement. (True)
- (e) When if statements are nested, the last else gets associated with the nearest if without an else. (false)
- (f) One if can have more than one else clause. (false)
- (g) Each expression in the else if must test the same variable. (True)
- (h) A switch statement can always be replaced by a series of if... else statements. (false)

3

- ① The predicate $!((x > 10) \wedge (y == 5))$ is equivalent to $(x < 10) \wedge (y \neq 5)$. (False)
- ② Any expression can be used for the if expression. (True)
- ③ It is mandatory to include an else block while using an if statement. (False)
- ④ The default block can be placed at the beginning of the switch case construct. (False)

Q2/ Fill in the blanks.

- ① The logical AND (&&) operator is true when both the operands are true.
- ② Multiway selection can be accomplished using an else if statement or the switch statement.
- ③ The break statement when executed in a switch statement causes immediate exit from the structure.

- 4
- ⑥ The expression $!(x \neq y)$ can be replaced by the expression $(x == y)$.
 - ⑦ The ternary conditional expression using the operator $?:$ could be easily coded using if...else statement.
 - ⑧ The if...else statement is used to branch unconditionally from one point to another point in a C program.

6.3 The program is,

```
n = 1;  
y = 1;  
if (n > 0)  
    n = n + 1;  
    y = y - 1;  
printf ("%d %d", n, y);
```

⑨ if n assumes a value of 1 then
 $n = 2, y = 0$

⑥ And n assumes a value of 0 then
 $n=1$ and $y=0$

6.9) Rewrite each of the following without using compound relations:

① if (grade ≤ 59 & grade ≥ 50)
second = second + 1;

Ans: if (grade ≤ 59) {
 if (grade ≥ 50) {
 second = second + 1;
 }
}

② if (number > 100)
 printf("out of range");
else if (number < 0)
 printf("out of range");
else
 printf("
 sum = sum + number;

6

```

c) if (M1 > 60) {
    if (M2 > 60)
        printf("Admitted\n");
    else if (T > 200)
        printf("Admitted\n");
}
else if (T > 200)
    printf("Admitted\n");
else
    printf("Not admitted\n");

```

6.5/ Assuming $x=10$;

- a) $x==10 \ \&\& \ x>10 \ \&\& \ !x$ (False)
- b) $x==10 \ || \ x>10 \ \&\& \ !x$ (True)
- c) $x==10 \ || \ x \leq 10 \ || \ !x$ (True)
- d) $x==10 \ \&\& \ x>10 \ || \ !x$ (False)

6.6/ Find errors:

a) correct ans: switch (x)

b) correct ans: case 10:

c) No error

d) correct ans: switch (x) { case 2: y = x + 2; break; }

6.7/ Simplify the following compound logical expressions.

a) $!(x < 10) = (x > 10)$

b) $!(x == 10) \parallel !(y == 5) \parallel (z < 0) = (x != 10) \parallel (y != 5) \&\& (z > 0)$

c) $!((x + y == z) \&\&!(z > 5))$

$= !(x + y == z) \parallel (z > 5)$

$= (x + y != z) \parallel (z > 5)$

d) $!((x < 5) \&\&(y == 10) \&\&(z < 5))$

$= (x > 5) \parallel (y != 10) \parallel (z >= 5)$ Ans

5.8

(a) if (x & y)

x = 10;

else

y = 10;

Output: x = 5; y = 10; z = 1

(b) if (x || y || z)

y = 10;

else

z = 0;

Output: x = 5; y = 10; z = 1

(c) if (x)

if (y)

z = 10;

else

z = 0;

Output: x = 5; y = 0; z = 0

(d) if (x == 0 || x & y)

if (y)

z = 0;

else

y = 1;

Output: x = 5; y = 0; z = 1

Q.5/

(a) switch (n)

{
 case 2:

 x = 1;

 y = x + 1;

 case 1:

 x = 0;

~~y = 0;~~

 break;

 default

 x = 1;

 y = 0;

Output: x = 1 ; y = 2

(Ans)

Q.10/ The output of the following program.

is 9. (Ans)

Q.11/ The output of the following

program is "Delhi", "Bangalore" "END"

6.12

```

main()
{
    int m;
    for (m=1; m<5; m++)
        printf("%d\n", (m%2)? m : m*2);
    getch();
}

```

Output: 2
4
6
8

6.13

The output of the following program is 0, 0, 2

6.14

```

int n=10; y=15;
n = (n < y) ? (y+n) : (y-n);
∴ n = 25

```


Q.15/ output:

Number is negative.

Q.16/ int a=10, b=5;

if (a>b)

{ if (b>5)

printf("%d", b);

}

else

printf("%d", a);

∴ output:- blank (Ans)

INTERVIEW QUESTIONS

Q.1/ In C programming language, there is no hard limit on the number of levels of nested blocks that can be created. The language specification does not define a maximum limit on the number of nested blocks that can be used in program.

However, the number of nested blocks that can be used in practice is limited by the available memory and the compiler's ability to handle deeply nested code. Deeply nested code can make the program more difficult to read, understand, and maintain, so it is generally considered good programming practice to limit the depth of nesting as much as possible.

6.2) If the conditional expression is missing in an if statement, the syntax of the statement will be invalid, and the program will likely fail to compile or execute.

The conditional expression in an if statement specifies the condition that is to be evaluated. The statement within the if block will only execute if the condition evaluates to true. Without a conditional expression, the compiler or interpreter does not have a condition to evaluate, and it cannot

determining whether the statement should be executed or not.

Q2 The 'goto' statement is a control flow statement in many programming languages that allows a program to jump to a different section of code based on a specified label. Its purpose is to provide an unconditional transfer of control to a specific point in the code.

The goto statement can be useful in certain situations, such as breaking out of nested loops or implementing error handling in code that doesn't support exceptions. However, its use is generally discouraged in modern programming practice because it can make code difficult to read and understand, and can lead to hard-to-debug problems such as infinite loops or code that jump around unpredictably.

6.9)

```
void main()
```

```
{
```

```
    int a=5;
```

```
    if (a < 0);
```

```
    printf("a is negative");
```

```
    else
```

```
    printf("a is positive");
```

```
}
```

output is "a is positive."

6.5/ In programming, an "if expression" is a conditional statement that evaluates a Boolean expression and executes a block of code if the expression is true. optionally, another block of code is executed if the expression is false.

2.61 In most programming languages, a switch statement can be used with values of a few specific types.

The most common types of values that are permitted in a switch statement are integers and characters. Some languages may also allow strings or enums to be used as switch cases.

In addition, some programming languages may allow the use of other types such as booleans or objects, but this is less common and may require special syntax or additional configuration.