Quiz-2 Detailed solutions

1. bool data type is best suited to represent logical values.

   bool → 0 & 1

2. Decision control statements in C++ are implemented using if, if else, ternary or conditional operators.

   cond ? ___ : ___ ;

   ↑ if ↑ else

3. 0 in C++ is treated as false & hence else block will be executed.

   Good Bye will be printed.

4. ```
   if (cond1) { ___ }
   else if (cond2) { ___ }
   else if (cond3) { ___ }
   ```

   Here by if else if we can check multiple conditions & then decide the flow of control.

5. do-while loop is executed atleast once whether the condition is false or true.
   ```

```
do {
  ---
} while (cond);
```

6. for
   while
   do-while
   } All loops have same speed

7. By using break; statement, we can come out from the loop.

```
while (cond) {
  ----
  break;   → Can be enforced on the basis of some particular condition.
  ---
}
```

8. i = 0 to i = 4 (i < 5)
   0, 1, 2, 3, 4 → 5 iterations

9. In switch (expr), expr should be an integer or character expression only.

10. We use colon (:) in the label of go to statements. It is the part of syntax.

11. n = 5 → print 5
5 == 3 (False) so no break

n = 4 → print 4
4 == 3 (False) so no break

n = 3 → print 3
3 == 3 (True) break

5 4 3 are printed on our screen.

12. Compile time error.

If label was used, then it would print 15 infinite number of times.

label : time

13. for (; ;) { → Treat this as ∞ loop
cout << n ; as there is no condition.
}
Hence infinitely prints the value of n.

14. This is a very good question & we need to have good observation to solve this question.

for (int i=0 ; i<10 ; i++) ; → This is imp
Here There is no body of for loop.

```
i = 10
cout << i ;
```
This prints 10 on our screen.

15.
```
a = 2
b = 7
c = (a > b) ? a : b ;
```

2 > 7 → False so b will be assigned to c.
Hence 7 is printed on the screen.

16.
```
a = 2 , b = 7
if ( 2 && 7) { -- }
```
2 and 7 are non-zero & hence are treated as true values.
Therefore true is printed on the screen.

17. | → Bitwise OR operator
|| → Logical OR operator

18.
```
c = a , b ;
```
a is assigned to c

```
d = (a, b) ;
```
o/p → b as , has left to right associativity

d = b

↱ value of a

5 6 is printed

↳ value of b

19. int i, j;

j = 10;

i = (j++, j+100, 999+j)

↳ j = 11 ↳ 100+11 = 111 ↳ 999+11 = 1010

Last value is assigned due to associativity being from left to right.

i = 1010 → printed on screen

20. int x, y;

x = 5;

y = ++x * ++x

7 x 7 = 49

x = 7, y = 49

cout << x << y; → 749

y = x++ * ++x

↓ ↓

7 9 = 63

x = 8 x = 9

x = 9, y = 63

cout << x << y; → 963

21. 21.09399 → int = 21 } Typecasting
10.20 → int = 10

22. a = 20
b = 10
c = 15
d = 5

e = a + b*c/d

Marking brackets according to precedence & associativity

e = (a + ((b * c) / d))

e = (20 + (150/5))

e = 20 + 30 = 50 Ans

23. From main(), execution of program will start.

24. int main() { }
↑ return type ↑ name } mandatory

Parameters are optional.

25. call by reference as copy won't be created & operations will be done on actual variables.

26. This gives us compile time error

No ( ) are used.

```
void car {

}
```

27. The x value remains unchanged & hence 10 is printed.

↳ Call by value concept

28. This gives an error as in the default parameters are always to the right but in this they are to the left of normal i/p parameters.

29. $5 + 0 = 5$. Hence 5 will be the o/p on the screen.

30. 1001

$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$8 + 0 + 0 + 1 = 9$ } decimal equivalent of 1001.

31. 13

$13/2 = 6$

$6/2 = 3$

$3/2 = 1$

$1/2 = 0$

1
0
1
1

1101 } Binary of 13

32. $15 \rightarrow 1111$

4 bits are required to represent 15 in binary format.

33. 1111

$1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$8 + 4 + 2 + 1 = 15$

34. int → 4 bytes = $4 \times 8 = 32$ bits

35. $a = 5$

$b = -7$

$c = 0$, $a = 6$, $b = -6$, $c = 0$, $d = 1$

$d = (((++a) \;\&\&\; (++b)) \;||\; (++c))$

$d = ((6 \;\&\&\; -6) \;||\; 0)$

↑ concept of shortcircuit

$d = 1 \;||\; 0$

$d = 1$

36. $a = -5$

$k = (a++ \,,\, ++a)\;;$

$k = -5 \,,\, -3$ (a = -4)

O/p = -3

37. ~0 → 9t is not 1 (discussed in notes)

Hence no is printed.

38. !0 = 1 → Hence yes is printed.

39. condition inside if is evaluated to true. y = 1 is also done. Hence the o/p is

y is 1.

40. Again condition inside if will be true. Value of y is also updated & y = 2.

Hence true 2 is printed. } However compiler dependent. Can be false 2 also.

41. a = 10, b = 5, c = 5

(d = ((b + c) == a))

d = (10 == 10)

d = 1

42. a = 10, b = 5, c = 3

(b! = (! a)) → (b! = 0) → 5! = 0

True = 1

But b remains 5 only

$(c = (!(!a)))$ → 3 == 0 → False = 0

!0 = 1

o/p is 5 1

**

43. Floating point number may be loose precision & hence else is executed. } B is print

**

44. x = 0.1

%d → Junk value as 0.1 is float

%f → 0.100000 — 6 is the precision

45. c = 2^3

000---- 010 ^ 000--- 011

= 000--- 001 = 1

46. a → 10

a = ~a

~ 0000---- 1010

1111--- 0101

2s complement ⇒ 0000-- 1010

1

0000-- 1011 → 11

→ negative number

**

47. sizeof is evaluated here

sizeof (x++); → x won't be updated

Hence o/p is (x is 97)

48. No break is used so when a case is matched, corresponding below cases will also be executed.

3 5 RABBIT RABBIT

From 1st switch 2nd switch o/p

49. Error. We don't have to use the case keyword with default.

50. MySwitch is printed.

51. HERO HONDA will be printed as - there is no case statement matching & also there is no default.

52. switch(a)(i) → This is imp to note

DEER LION is printed.

53. 64+1 = 65 ASCII value of 'A'.
Hence ANT is printed. ANT PALACE

54. No case was matched. So default case is executed.

FLOWER GARDEN
From switch.

55. Compile time error as float can not be used in case of switch.