

Data for Q10 and Q11 are given below.

Solve the problems and choose correct.

answer

Consider the following program

```
int x = 10;
main ()
{
    p(x);
    print (x);
}
p (int a)
{
    if ( a ≤ 40)
    {
        a = a+10;
        p(a);
        x = a + 10;
    }
}
```

10. What value is printed if the program uses “Call by Value”? *(Answer: 10)*

11. What value is printed if the program uses “Call by Reference”?

For Que. 10 \times (11, 12)

int $x = 10;$

main()

{

p(x);

printf(x);

}

p(int a)

{

if ($a \leq 40$)

{

$a = a + 10;$

p(a);

$x = a + 10;$

}

}

10) Call by Value \rightarrow

?

11) \rightarrow

Call by reference \rightarrow

?

(a) $L_0 \Rightarrow$

Call by value —

$$x = 10;$$

(1) $\rightarrow p(10)$

{ if ($10 \leq 40$) }

$$a = 10 + 10 \Rightarrow a = 20 \leftarrow$$

(2) $\rightarrow p(20)$

$$\{ \{ a = 20 + 10 \Rightarrow a = 30 \} \leftarrow$$

(3) $\rightarrow p(30)$

$$\{ \{ a = 30 + 10 \Rightarrow a = 40 \} \leftarrow$$

(4) $\rightarrow p(40)$

$$\{ \{ a = 40 + 10 \Rightarrow a = 50 \} \leftarrow$$

(5) $\rightarrow p(50)$

{ if ($50 \leq 40$) } \rightarrow False

Retrive — $a = 40$

$$x = a + 10 = 40 + 10 = 50$$

Retrive — $a = 30$

$$x = 30 + 10 = 40$$

Retrive — $a = 20$

$$x = 20 + 10 = \boxed{30} \text{ Ans}$$

Sol \rightarrow

(a)

$$x = 30$$

12. Macro processing is done during
- (a) compile time
 - (b) load time
 - (c) runtime
 - (d) link time

(a) 128

Macro processing is done during compile time.

Because — Compiler ~~replaces~~ replaces the ~~macro~~ values at the time of compilation.

Ans → (a) Compile Time

13. _____ is a num type for which we can define the operations to be used for manipulating instances, while the data structure that implements the type is hidden to the users.

- (a) Primitive data type
- (b) Based-data type
- (c) Abstract data type
- (d) Object-data type

(c) \rightarrow Abstract-data-type is a num type for which we can define the operations to be used for manipulating instances, while data-structure that implements the type is hidden to the user.

Ans \rightarrow (c) Abstract Data Type

14. Which of the following is *not* a valid declaration for main()?

- (a) int main()
- (b) int main (char * arg c[], int arg v)
- (c) int main (int arg v, char * arg v [])
- (d) a, b & c are correct

d 14s

declaration of main() in C →

(i) int main() { }

(ii) int main (char*arg c[], int arg v) { }

(iii) int main(char int arg v, char*arg v[]) { }

d

All declarations are correct.

15. After a function has returned, the values stored in its local variables are copied into variables in the main function with the same name

① b) 15 →

int f1()

int a = 10; // local variable

Local variable life time ends when function
end.

Ans 3 ⑥ False

mean memory addressing in optimization

16. Consider the following statements:

- I. The l – value of a variable is the storage area bound to the variable during execution.
- II. The r – value of a variable is the encoded value stored in the location associated with the variable.
- III. Some languages allow variables that can be accessed through the r – value of another variable known as reference.

Which of the above statements are *false*?

- (a) I & II
- (b) II & III
- (c) I, II & III
- (d) none of the above

Q16

Statements :-

True

(i) The l-value (left-value) of a variable is the storage area bound to the variable during execution.

True

(ii) The r-value (right-value) of a variable is the encoded value stored in the location associated with the variable.

True

(iii) Some languages allow variables that can be accessed through the r-value of another variable known as reference.

⇒ "l-value" may appear as either LHS or RHS of an assignment operator ("=").
"l-value" often represents an identifier.

Example ↳

int a; // a is an object, type - int
a = 1; // a is l-value on left-hand-side

int b = a; // Now, a on right hand side

But $\theta = a;$ // Compile-Time Error

⇒ "r-value" can appear on right but not left-hand-side of an assignment operator (=)

Because "r-value" is an expression.

Example →

int a = 1, b; // Here a, b are objects
a + 1 = b; // Error

17. Match the following:

- | | |
|------------|----------------------|
| (1) C | (M) Imperative |
| (2) Java | (N) Functional |
| (3) Lisp | (O) Object oriented |
| (4) Prolog | (P) Constraint based |

Codes:

- (a) 1 - M, 2 - N, 3 - O, 4 - P
- (b) 1 - P, 2 - O, 3 - M, 4 - N
- (c) 1 - M, 2 - P, 3 - O, 4 - N
- (d) 1 - M, 2 - O, 3 - N, 4 - P

Q17

Match Enter (do something - do something)

- | | | |
|------------|----|----------------------|
| (1) C | is | (M) Imperative |
| (2) Java | is | (N) Functional |
| (3) Lisp | is | (O) Object Oriented |
| (4) Prolog | is | (P) Constraint based |

Ans: 1 → M, 2 → O, 3 → N, 4 → P

18. Consider the following 'C' statement. What is the order in which functions are called?

$a = f1(23, 14) * f2(12/4) + f3();$

- (a) f1, f2, f3
- (b) f3, f2, f1
- (c) The order may vary from compiler to compiler
- (d) None of the above

(C) 183

Function () → ?

$$a = f1(23, 14) * f2(12/4) + f3();$$

Ans In "C" language has many compilers.

So -- order of functions may vary from compiler to compiler.

19. In which of the following parameter passing mechanisms, the actual argument has to be a variable.

- (a) call by value
- (b) call by name
- (c) call by reference
- (d) call – by – need

Q 19)

Which parameters passing mechanism, actual argument has to be a variable - ?

Ans \Rightarrow Call - by - reference

Call - by - reference accept address , off the variable ,
actual argument must be variable.

Teacher's Signature