|  |  |
| --- | --- |
| **Week 4 Task** | **Title: Control statements** |
| **Name and Student ID** |  |

**Tutorial Task(s)**

1. Explain the keyword: a) Block b) Boolen expression c) Loop
2. What do you mean by control statement?
3. Explain the purpose of if-else statement?
4. Explain the purpose of switch statement?
5. What is the purpose of comma operator?
6. Compare between a pre-test and post test loop?
7. Mention the reason “ why is the use of goto statement discouraged?
8. Is the relational expression a < b < c legal in C?
9. There is no logical exclusive OR operator in c program, can it it be simulated?
10. The floating point numbers are seldom equals to required value of a variable, then how can floating point values or variables be tested for equality?
11. Explain null statement?
12. Which form of loop should use – while or do-while?
13. Differentiate between break and continue statement
14. Is there any difference between the following for statement? Explain?
15. for(x = 1; x < 100; x++)
16. for(x =1; x < 100; ++x)
17. for (x = 1; x < 100; x = x + 1)
18. for (x = 1; x< 100; x += 1)
19. What would be the output of following program?

int main()

{

int i=9;

for(i--; i--; i--)

printf(“%d”, i)

return 0;

}

1. What will be printed by the code below?

a)

float x = 123.4

If ( x< 100)

printf (“one “);

If ( x < 200)

printf ( “ two “);

If (x < 300)

printf( “ three “);

b)

char c = ‘y’;

switch(c)

{

case ‘Y’; printf(“Yes/No”);

break;

case ‘N’; printf(“No/ Yes”);

break;

default: printf(“other”);

}

Practice!!!!!

Try the below project-based questions:

1. Write a C program that prompts the user to enter the date as three integer values for the month, the day in the month and year. The program should then output the data in the form of 31stDecember 2003 when the user enters 12 31 2010, say. The program has to work out when c “th”, “nd”, “st”, and “rd” need to be appended to the day value. The programmer should not forget 1St, 2nd , 3rd, 4th; and then 11th , 12th, 13th, 14th; and 21st, 22nd, 23rd, and 24th.
2. This is a well-known game with number of variants. The following variants has an interesting winning strategy. Two players alternatively take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then the other player takes a turn. The player who takes the lost marble loses. Write a C program in which computer plays against a human opponent. Generate a random integer between 10 and 100 to denote the initial size of the pile. Generate a random integer between 0 and 1 to decide whether the computer or human takes the first turn, generate a random integer between 0 and 1 to decide whether computer play smart or stupid. In the stupid mode, the computer simply takes a random legal value ( between 1 and n/2, where n is the total number of marbles) from the piles whenever it has a turn. In smart mode, the computer takes off enough marbles to make the size of the piles a power of two minus- that is, 3, 7, 15, 31 or 63. That is always a legal move, except when the size of pile is currently one less than power of two. In that case, the computer makes a random legal move. It should be noted that the computer cannot be beaten in smart mode when it has the first move, unless the pile size happens to be 15, 31 or 63. Of course a human player who has the first turn and knows the winning strategy can win against the computer.