GLS UNIVERSITY

FACULTY OF COMPUTER APPLICATIONS & IT

SUBJECT: 0301201 Introduction to Object Oriented Programming

BCA Sem – II

Theory Assignment – II

Date of Assignment: 28/12/2022

Note: Student can attach print out for all the questions except question 5.

Take Print out and fill answer in it and submit to corresponding faculty.

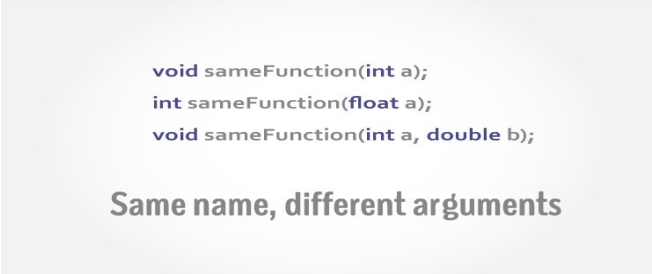
Q-1 Fill in the Blanks

1. _____ type of operator defined with a single operand.
2. _____ looping is executed atleast at once even though condition is false.
3. A relational expression is also called _____ expression or _____ expression.
4. A set of statements enclosed within a pair of opening and closing braces is called _____ statement.
5. A _____ statement provides an unconditional jump.
6. _____ type of operator is defined with two operands.
7. _____ is used when block of code is to be executed n number of times.
8. _____ structure/statement is an alternative to else-if ladder which simpli fies the code and enhances readibility.
9. If some operands are of integer type and some are of float type in an expression then it is called _____ mode expression.
10. The _____ is a keyword and operator that determines the size, in bytes, of a variable or data type.
11. _____ type of operators is defined with three operands.
12. Conditional operator is also known as _____.
13. Function header + Function body = _____
14. The first line consists of return type, function name and arguments declaration enclosed within a pair of parentheses is called _____
15. Compile time polymorphism is also called early binding or _____
16. A _____ variable is an alias to an existing variable.
17. _____ operator resolves dispute in local and global variables with same name.
18. The signature of a function is also known as _____

19. To inline a function, place the keyword _____ before the function name and define the function before any calls are made to the function.

20. C++ allows you to specify more than one definition for a function name in the same scope, which is called _____

21. Identify the picture and state the name of concept:



22. There are two ways to overload the method in C++

    i By changing _____ of arguments or parameters

    ii By changing the _____

23. A _____ is a value provided in function declaration that is automatically assigned by the compiler if caller of the function doesn't provide a value for the argument with default value.

24. A function that calls itself is known as recursive function. And, this technique is known as _____.

25. ____ datatype is used to represent the absence of parameters.

26. Function declaration statement must end with _____.

27. Inline function is used when _____.

28. default value for parameters is assigned in _____

Q-2 True or False

1. In switch case default case is optional.
2. The comma operator (,) is used to separate two or more expressions.
3. Break statement forces the next iteration of the loop to take place, skipping any code in between.
4. If block is always followed with the optional else block.
5. While loop test the condition at the end of the loop only.
6. Looping is also called iteration.
7. Do... while loop is followed by three fundamentals things to construct loop, initialization and test-expression.
8. A while loop can be nested inside another loop.

9. Remainder Operators (%) are not compatible with floating point numbers.
10. The do-while loop is similar to the while loop, except that the test condition occurs at the end of the loop.
11. A function which invokes another function is called calling function.
12. A function returns float value by default.
13. Function overloading is also termed as function polymorphism.
14. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.
15. The compiler can ignore the inline qualifier in case defined function is more than a line.
16. Compiler can ignore the request for in lining, If a function is recursive.
17. Compiler can ignore the request for in lining, If a function contains switch or goto statement.
18. Compiler can ignore the request for in lining, If a function return type is other than void, and the return statement doesn't exist in function body.
19. Compiler can ignore the request for in lining, If a function contains static variables.
20. For inline function, Function call overhead doesn't occur.
21. Inline function also saves the overhead of push/pop variables on the stack when function is called.
22. An overloaded declaration is a declaration that had been declared with the same name as a previously declared declaration in the same scope, except that both declarations have different arguments and obviously different definition (implementation).
23. State whether following definitions are valid for function overloading or not.
24. int test() { } int test(int a) { } float test(double a) { } int test(int a, double b) { }
25. For the default argument concept only the last argument must be given default value.
26. default argument cannot be followed by non-default argument.
27. State whether following declarations for default argument are valid or not.
28. sum (int x,int y=0);

    sum (int x,int y=0,int z);

    sum (int x,int y=10,int z=10);

    void sum (int, int=0);

void add(int a, int b = 3, int c, int d = 4);

void add(int a, int b = 3, int c, int d);

add(int a, int b, int c, int d = 4);

29. One can give any value a default value to argument, compatible with its datatype.

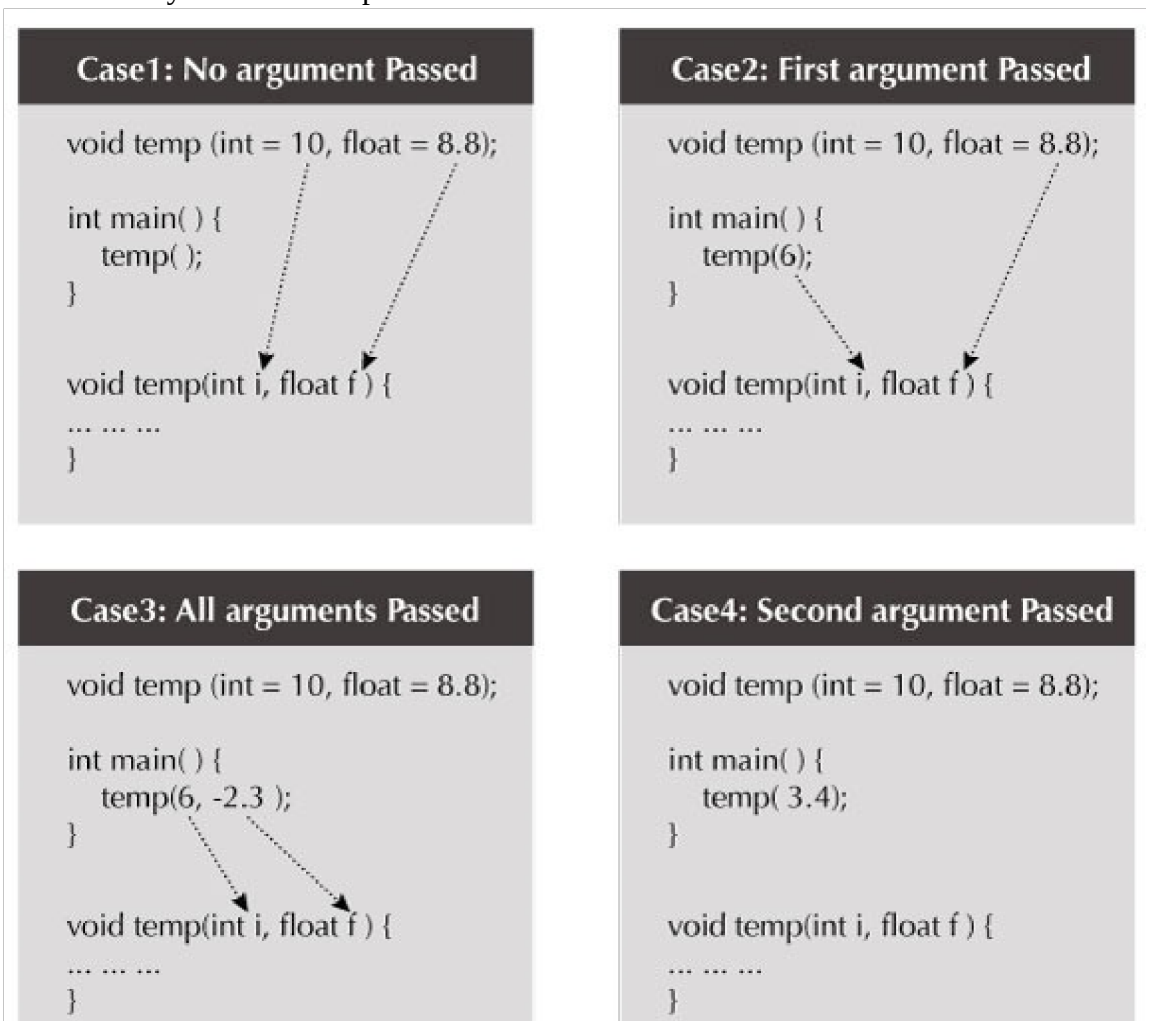
30. When arguments in a function are declared without any identifier they are called placeholder arguments.

31. Can Function return array.

32. Column size must be passed in function declaration, when passing double dimension array as an argument.

33. The idea behind default argument is simple. If a function is called by passing argument/s, those arguments are used by the function. But if the argument/s are not passed while invoking a function then, the default values are used.

34. Verify case 4 and explain:



Q-3 Define following keywords/functions.

1. goto
2. break
3. continue
4. sizeof()
5. stoi()
6. stod()
7. else
8. default
9. Nested if
10. inline
11. Default Argument
12. Function Overloading
13. Recursion
14. Return statement
15. function

Q-4 Find the Output:

1.
```
#include<iostream>
using namespace std;
int main()
{
int i;
for(i=1;i<=10;i++);
{
        cout<<i;
}
        return 0;
}
```

2.
```
#include<iostream>
  int main() {
       float x=5,y=2;
```

```
                int result;
                result=x % y;
                cout<<result;
            }
```

3.
```
    #include<iostream>
      void main()
        {
          float x;   x=(float)9/2;   cout<<x;
        }
```

4.
```
    #include<iostream>
using namespace std;
int main() {
int x=10, y=1, z;
z=++x+y+++--x+--y;
cout<<"\n z is:"<<z;
return 0;
}
```

5.
```
    #include<iostream>
using namespace std;
int main()
{
        long double x=100;
        cout<<"\n x is:"<<x;
        cout<<"\n address of x is:"<<(void *)&x;
        cout<<"\n size of x is:"<<sizeof(x);
        return 0;
}
```

6.
```
    #include<iostream>
        using namespace std;
        int main()
{
        int a=10; int b,c=5,d=20,f=1000,j=200;
```

```
        b=c=a+20; a=b*c*d-f/j+10;
        cout<<a;
    }
```

7.
```
    #include<iostream>
    using namespace std;
    int main()
    {
        char var='A'; switch(var)
        {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U':
        cout<<"\n CHAR IS VOWEL:";
        break;
        default:
        cout<<"\n CHAR IS CONSONENT:";
    }
    return 0;
}
```

8.
```
    #include<iostream>
    using namespace std;
    int main()
    {
    int a=10;
    if(a=5)
```

```
{
        cout<<"\n"<<a;
}
else
{
        cout<<"not executed";
}
        return 0;
}
```

9.
```
#include<iostream>
using namespace std;
int main()
{
        a=10; c=5; b=c,c=a+20;
        cout<<"\n"<<a;
        cout<<"\n"<<b;
        cout<<"\n"<<c;
        return 0;
}
```

10.
```
#include<iostream>
  using namespace std;
  int main() {
    int x=10, y=-1, z;
    z=++x+y+++--x+--y;
    cout<<"\n z is:"<<z;
    return 0;
    }
```

Q-5 Answer the following questions:

1. What are Bitwise operators?explain with the help of example.
2. Explain the concept of operators. Explain term expressions and operands.
3. Explain nested if else structure with example.
4. Explain for loop with its syntax.
5. Differentiate break and continue statement.
6. What are Assignment operators?explain with the help of example.
7. Differentiate if-else ladder with switch statement.
8. What do you mean by term precedence and associativity?
9. Explain do-while loop with example.
10. What are binary operators?explain with the help of example.
11. Differentiate while and do..while loop
12. Explain for loop with example.
13. Explain the working of logical && and logical || operator.
14. Explain the concept of goto statement.
15. Explain comma operator with the help of example.
16. Explain Shorthand Arithmetic Assignment Operators with example.
17. Explain Ternary operator with the help of example.
18. What is Function? Explain advantages of functions.
19. List and explain classification of function in detail.
20. Differentiate built-in function and user-define function with example.
21. What do you mean by function header and function call. Explain.
22. Differentiate calling function and called function.
23. Explain the concept of recursion with example.
24. What is inline function? Explain its significance.
25. Explain the concept of Function Overloading with example.
26. Explain the concept of default argument in C++.

Note: All the students have to attempt Q-1, Q-2, Q-3 and Q-4 compulsory.

Attempt Q-5 in following sequence:

Roll No.	**Question No.**
A01 to A15, B01 to B15, C01 to C15,AS01 to AS15, BS01 to BS15	1,9,17,25
A16 to A30, B16 to B30, C16 to C30,AS16 to AS30, BS16 to BS30	2,10,18,26
A31 to A45, B31 to B45, C31 to C45,AS31 to AS45, BS31 to BS45	3,11,19,1
A46 to A60, B46 to B60, C46 to C60,AS46 to AS60, BS46 to BS60	4,12,20,2
A61 to A75, B61 to B75, C61 to C75,AS61 to AS75, BS61 to BS75	5,13,21,3
A76 to A90, B76 to B90, C76 to C90,AS76 to AS90, BS76 to BS90	6,14,22,4
A91 to A105, B91 to B105, C91 to C105,AS91 to AS105, BS91 to BS105	7,15,23,5
A106 onwards, B106 onwards,C106 onwards,AS106 onwards, BS106 onwards	8,16,24,6