SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956) THINK MERIT | THINK TRANSPARENCY | THINK SASTRA	**School of Computing** **First CIA Examination – Sep 2023** Course Code: CSE213 Course Name: Object Oriented Programming Duration: 90 minutes Max Marks: 50

PART A (2x10=20)

Answer all the questions

1. Distinguish between structures and classes.

Structure	Class
Classes have access modifiers like public, private, and protected	Structures do not support access modifiers
Classes support inheritance	Structures do not support inheritance.
Classes often have default constructors, destructors and other special methods generated by the compiler	Structures typically do not have these default methods generated by the compiler
Memory for class objects is allocated on the heap.	Memory for structure objects is allocated on the stack

2. Fill in the blank with Trigraph Sequences

```
___define MSG "Hello"
___define Program "CSBS"
int main()
___
        cout << "My message to " << MSG << endl;
        cout << "My program is " << Program;
        return 0;
___
```

```
??=define MSG "Hello"
??=define Program "CSBS"
int main()
??<
   cout << "My message to " ??( MSG ??) << endl;
   cout << "My program is " ??( Program ??);
   return 0;
??>
```

3. List the various methods of passing arguments to a function with example.
   Pass by Value
   Pass by Reference
4. Define constructor and list its type.
   A constructor in object-oriented programming is a special method or function that is used to initialize and set up the initial state of an object when an instance of a class is created. It typically has the same name as the class it belongs to and is called automatically when an object is instantiated. Constructors are used to allocate memory for the object and initialize its attributes or properties.
   Types: Default Constructor
   Parameterized Constructor
   Copy Constructor
5. Write the building blocks of Exception handling routine.
   Try, Catch, Throw block
6. What is dangling pointer and memory leak?
   a. A dangling pointer is a pointer that continues to reference a memory location after the memory it points to has been deallocated or released.
   b. A memory leak occurs when a program allocates memory from the system but fails to release or deallocate that memory
7. Write short note on Enumeration data type with example.
   "enum," is a user-defined data type in programming that consists of a finite set of named values. Enumerations are used to represent a set of constant, related values with descriptive names,
8. Write the suitable technique to reduce function call overhead and justify it.
   Inline Function

When a function is inlined, the compiler replaces the function call with the actual code of the function, making the program execution faster.

9. State the difference between #pragma startup and #pragma exit directives in terms of program execution.

   #pragma startup and #pragma exit are compiler-specific directives that allow you to specify functions to be executed at the start and the end of a program's execution, respectively.

10. Predict the output for the code

```
#include <iostream>
 #include <iomanip>
using namespace std;
int main()
{
 cout << setfill('$') << setw(5);
cout << "10" << endl;
}
```

Output: $$$10

PART B (3x10=30)

Answer All the Three Questions

11. Consider a class named Calculator that contains overloaded functions for calculating areas of different geometric shapes. Implement the class with appropriate member functions to calculate the area of a rectangle, a circle, and a triangle.

    Hint: The class should have public member functions named calculateArea for each of the following shapes:

    Rectangle: calculateArea(double length, double width)

    Circle: calculateArea(double radius)

    Triangle: calculateArea(double base, double height)

    Answer:

    - Write code that has the Calculator class contains three public member functions named calculateArea, each taking the required parameters for the respective geometric shape (rectangle, circle, or triangle).
        - The first calculateArea function calculates the area of a rectangle given its length and width.
        - The second calculateArea function calculates the area of a circle given its radius. It uses the constant M_PI from the <cmath> library for pi (π).
        - The third calculateArea function calculates the area of a triangle given its base and height.
        - In the main function, an instance of the Calculator class is created, and the calculateArea functions are called to calculate and display the areas of a rectangle, a circle, and a triangle.

12. Define a symbolic constant called STATE, set to initial value TAMILNADU. In the main() function, check if STATE is defined;

    if yes, then
    check if its value is TAMILNADU;
    If YES, then
        define another symbolic constant as CAPITAL with its value set as Chennai.
    Else
        Print the message “Name wrongly set”
    Else
        Define a symbolic constant called STATE and set its value as GUJARAT.

    Implement the above using appropriate pre-processor directives and include test cases to verify all possible outputs.

Ans:

```
#include <stdio.h>
#define TAMILNADU 1
#define GUJARAT 2
int main() {
  #ifdef STATE
    #if STATE == TAMILNADU
      #ifdef CAPITAL
        printf("State: TAMILNADU\n");
        printf("Capital: Chennai\n");
      #else
```

```
            printf("State: TAMILNADU\n");
            printf("Name wrongly set\n");
        #endif
    #endif
#else
    #define STATE GUJARAT
    #define CAPITAL "Ahmedabad"
    printf("State: GUJARAT\n");
    printf("Capital: %s\n", CAPITAL);
#endif
return 0;
}
```

13. Create a class called employee that contains a name (an object of class string) and an employee number (type long). Include a member function called getdata() to get data from the user for insertion into the object, and another function called putdata() to display the data. Assume the name has no embedded blanks. Write a main() program to exercise this class. It should create an array of type employee, and then invite the user to input data for up to 100 employees. Finally, it should print out the data for all the employees.

```
#include <iostream>
#include <string>

using namespace std;

class Employee {
private:
    string name;
    long employeeNumber;

public:
    void getdata() {
        cout << "Enter Employee Name: ";
        cin >> name;
        cout << "Enter Employee Number: ";
        cin >> employeeNumber;
    }

    void putdata() {
        cout << "Employee Name: " << name << endl;
        cout << "Employee Number: " << employeeNumber << endl;
    }
};

int main() {
    int numEmployees;

    cout << "Enter the number of employees (up to 100): ";
    cin >> numEmployees;

    if (numEmployees > 100 || numEmployees <= 0) {
        cout << "Invalid number of employees. Please enter a value between 1 and 100." << endl;
        return 1;
    }

    Employee employees[100]; // Assuming a maximum of 100 employees

    for (int i = 0; i < numEmployees; ++i) {
        cout << "Enter details for Employee " << i + 1 << ":" << endl;
        employees[i].getdata();
    }

    cout << "Employee Details:" << endl;
    for (int i = 0; i < numEmployees; ++i) {
        cout << "Employee " << i + 1 << ":" << endl;
        employees[i].putdata();
        cout << endl;
    }

    return 0;
}
```