**Roll No…………….. Total No. of Pages:……**

**ST-2 (SET-V )**

**4th SEMESTER 2022-23**

**22CS006- Object Oriented Programming**

**Time allowed: 90 Minutes Max. Marks: 40**

**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1. How is the "this" pointer represented in C++?
   1. this.ptr
   2. self
   3. thisptr
   4. **this**
2. What is the purpose of the "new" operator in C++?
   1. **Allocate memory for a new object**
   2. Create a new class definition
   3. Allocate memory for a variable
   4. Create a new object with initial values
3. What is the keyword used to destroy an object explicitly in C++?
   1. break
   2. **delete**
   3. free
   4. destroy
4. Which operator is used to access class members through a pointer to an object?
   1. \*
   2. **->**
   3. ::
   4. .
5. What is the function of a static member variable in a class?
   1. It is a constant value for all objects of the class
   2. **It is shared among all objects of the class**
   3. It is accessible only within the class
   4. It is used to control access to private members
6. Which of the following is true about the default constructor?
   1. **It is automatically provided by the compiler if no constructors are defined**
   2. It is automatically called when an object is destroyed
   3. It is automatically called when an object is created
   4. It is automatically provided by the compiler if the class has a destructor
7. When does the copy constructor get called in C++?
   1. When an object is passed to a function by value
   2. When an object is returned from a function by value
   3. When an object is explicitly destroyed using the delete keyword
   4. **When an object is created using another object of the same class**
8. What is the purpose of using the "explicit" keyword with a constructor?
   1. **It prevents the constructor from being called implicitly**
   2. It allows the constructor to be called implicitly
   3. It makes the constructor a friend of the class
   4. It converts the constructor into a virtual constructor
9. Can a destructor be overloaded in C++?
   1. Yes
   2. **No**
10. Which constructor is called first in a derived class?
    1. Derived class's constructor
    2. **Base class's constructor**
    3. Destructor
    4. Copy constructor

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

1. What is the output of the following code?

#include <iostream>

using namespace std;

class Base {

public:

Base() { cout << "Base constructor" << endl; }

virtual ~Base() { cout << "Base destructor" << endl; }

};

class Derived : public Base {

public:

Derived() { cout << "Derived constructor" << endl; }

~Derived() { cout << "Derived destructor" << endl; }

};

int main() {

Base\* ptr = new Derived();

delete ptr;

return 0;

}

1. **Base constructor, Derived constructor, Derived destructor, Base destructor**
2. Derived constructor, Base constructor, Base destructor, Derived destructor
3. Base constructor, Derived constructor, Base destructor
4. Derived constructor, Base destructor
5. What will be the output of the following C++ code?

#include <iostream>

using namespace std;

class CDummy

{

public:

int isitme (CDummy& param);

};

int CDummy::isitme (CDummy& param)

{

if (&param == this)

return true;

else

return false;

}

int main ()

{

CDummy a;

CDummy \*b = &a;

if (b->isitme(a))

{

cout << "execute";

}

else

{

cout<<"not execute";

}

return 0;

}

1. **execute \*\*\*\*\*\*\***
2. not execute
3. error
4. both execute & not execute
5. What is the correct syntax of accessing a static member of a Class?

Example class:

class A

{

public:

static int value;

}

* 1. A.value
  2. **A::value**
  3. A->value
  4. A^value

1. What will be the output of the following C++ code?

#include <iostream>

using namespace std;

class S

{

int m;

public:

#define MAC(S::m)

};

int main(int argc, char const \*argv[])

{

cout<<"Hello World";

return 0;

}

* 1. Hello World
  2. **Error**
  3. Segmentation Fault
  4. Blank Space

1. What will be the output of the following C++ code?

#include <iostream>

#include <string>

using namespace std;

class A

{

static int a;

public:

void show()

{

a++;

cout<<"a: "<<a<<endl;

}

};

int A::a = 5;

int main(int argc, char const \*argv[])

{

A a;

return 0;

}

* 1. Error as a private member a is referenced outside the class
  2. Segmentation fault
  3. **No output**
  4. Program compiles successfully but gives run-time error

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16) You have given a string. Write a function that reverses a string using a stack data structure.

**Input :** hello

**Output:** olleh

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | JAVA | Stack Overflow | Pune |
| **Output** | AVAJ | wolfrevO kcatS | enuP |

Solution :

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_SIZE 100**

**char stack[MAX\_SIZE];**

**int top = -1;**

**void push(char ch) {**

**//check for stack is full or not**

**if (top == MAX\_SIZE - 1) {**

**printf("Stack Overflow\n");**

**return;**

**}**

**//push element to stack**

**stack[++top] = ch;**

**}**

**char pop() {**

**//check for stack is empty or not**

**if (top == -1) {**

**printf("Stack Underflow\n");**

**return -1;**

**}**

**//remove element from top of the stack**

**return stack[top--];**

**}**

**void reverseString(char\* str) {**

**int len = strlen(str);**

**//push all characters of the string one by one**

**for (int i = 0; i < len; i++)**

**push(str[i]);**

**//pop all characters of the string one by one and store in same string**

**for (int i = 0; i < len; i++)**

**str[i] = pop();**

**//after this, str will contain reversed string**

**}**

**int main() {**

**char str[100];**

**printf("Enter a string: ");**

**scanf("%[^\n]%\*c", str);**

**printf("Original string: %s\n", str);**

**reverseString(str);**

**printf("Reversed string: %s\n", str);**

**return 0;**

**}**

Q17) You are tasked with developing a program that performs number swapping using macros.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 5 10 | 18 93 | 9 7 |
| **Output** | Before swap: x = 5, y = 10  After swap: x = 10, y = 5 | Before swap: x = 18, y = 93  After swap: x = 93, y = 18 | Before swap: x = 9, y = 7  After swap: x = 7, y = 9 |

Solution :

**#include <stdio.h>**

**#define SWAP(a, b) do { \**

**int temp = (a); \**

**(a) = (b); \**

**(b) = temp; \**

**} while (0)**

**int main() {**

**int x = 5, y = 10;**

**printf("Before swap: x = %d, y = %d\n", x, y);**

**SWAP(x, y); //cal SWAP Macro**

**printf("After swap: x = %d, y = %d\n", x, y);**

**return 0;**

**}**

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18) You are given an array of integers nums, there is a sliding window of size k which is moving from the very left of

the array to the very right. You can only see the k numbers in the window. Each time the sliding window moves right by one position.

Return the max sliding window.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | [1,3,-1,-3,5,3,6,7], k = 3 | [2, 3, 0, -6, 4, 3], k = 3 | [2, 3, 0, -6, 4, 3], k = 2 |
| **Output** | [3,3,5,5,6,7] | [3,3,4,4] | [3,3,0,4,4] |

Solution :

**#include<stdio.h>**

**#include<stdlib.h>**

**#define MAX\_SIZE 100**

**int adjMatrix[MAX\_SIZE][MAX\_SIZE];**

**int visited[MAX\_SIZE];**

**struct queue**

**{**

**int size;**

**int f;**

**int r;**

**int\* arr;**

**};**

**//check if queue is empty**

**int isEmpty(struct queue \*q){**

**if(q->r==q->f){**

**return 1;**

**}**

**return 0;**

**}**

**//check if queue is full**

**int isFull(struct queue \*q){**

**if(q->r==q->size-1){**

**return 1;**

**}**

**return 0;**

**}**

**//add element in queue**

**void enqueue(struct queue \*q, int val){**

**if(isFull(q)){**

**printf("This Queue is full\n");**

**}**

**else{**

**q->r++;**

**q->arr[q->r] = val;**

**// printf("Enqued element: %d\n", val);**

**}**

**}**

**//remove element from queue**

**int dequeue(struct queue \*q){**

**int a = -1;**

**if(isEmpty(q)){**

**printf("This Queue is empty\n");**

**}**

**else{**

**q->f++;**

**a = q->arr[q->f];**

**}**

**return a;**

**}**

**void BFS(int start,int numVertices){**

**// Initializing Queue (Array Implementation)**

**struct queue q;**

**q.size = 400;**

**q.f = q.r = 0;**

**q.arr = (int\*) malloc(q.size\*sizeof(int));**

**// BFS Implementation**

**int node;**

**printf("%d ", start);**

**visited[start] = 1;**

**enqueue(&q, start); // Enqueue i for exploration**

**while (!isEmpty(&q))**

**{**

**int node = dequeue(&q);**

**for (int j = 0; j < numVertices; j++)**

**{**

**if(adjMatrix[node][j] ==1 && visited[j] == 0){**

**printf("%d ", j);**

**visited[j] = 1;**

**enqueue(&q, j);**

**}**

**}**

**}**

**}**

**int main(){**

**int numVertices;**

**printf("Enter the number of vertices: ");**

**scanf("%d", &numVertices);**

**printf("Enter the adjacency matrix:\n");**

**for (int i = 0; i < numVertices; i++) {**

**for (int j = 0; j < numVertices; j++) {**

**scanf("%d", &adjMatrix[i][j]);**

**}**

**}**

**// Initialize visited array**

**for (int i = 0; i < numVertices; i++) {**

**visited[i] = 0;**

**}**

**int startVertex;**

**printf("Enter the starting vertex: ");**

**scanf("%d", &startVertex);**

**BFS(startVertex, numVertices);**

**return 0;**

**}**