

Output :-

array after sorted

001112222

B. Write a program to sort an array 0, 1, 2 in the best possible time and space complexity.

```
#include <bits/stdc++.h>
using namespace std;
void sort(int a[], int arr-size)
{
    int lower = 0;
    int high = arr-size-1;
    int mid = 0;
    while (mid <= high) {
        switch (a[mid]) {
            case 0:
                swap(a[lower++], a[mid++]);
                break;
            case 1:
                mid++;
                break;
        }
    }
    void printArray(int arr[], int arr-size, int i) {
        cout << arr[i] << " ";
    }
    int main ()
    {
        int arr[] = {1, 1, 2, 2, 0, 0, 2, 1, 2, 2};
        int n = sizeof(arr) / sizeof(arr[0]);
        sort(arr, n);
        cout << "array after segregation";
        printArray(arr, n);
        return 0;
    }
}
```


<pre>int main () { ComplexNum c1(10, 2), c2(3, 7); ComplexNum c3 = c1 + c2; c3.print(); }</pre>	<p>Output:</p> <p>The int is 5</p> <p>The float is 5.5.</p> <p>The char is five.</p>
---	--

2. Run Time polymorphism: In this functions are called at the time the program execution. Hence it is known as late binding or dynamic binding. Function overriding is a part of this.

Ex:- #include <iostream>
using namespace std;
class mammal {

public:

void eat() {

cout << "mammals eat";

}

};

class Cow: public Mammal {

public:

void eat() {

cout << "Cows eat grass";

}

Cows eat grass.

}

};

int main (void) {

Cow c = Cow();

c.eat();

return 0;

}

3. Copy Constructor: These are special type of constructor which takes an object as argument, and is used to copy values of data members of one object into the object.

Syntax:-

classname (const classname & objectname)

3. Explain the difference between object oriented and procedural programming language in detail.

Ans: POP

OOP

1. In POP, program is divided into small parts called functions.

In OOP, program is divided into small parts called objects.

2. It follows top-down approach.

It follows bottom-up approach.

3. There is no access specifier in this

This have access specifiers like private, public, protected

4. It is less secure

It is more secure

5. It is based on unreal world.

It is based on real world.

Ex: C, Pascal, Basic

Ex: C++, Java, Python

Long answer type Questions:-

A). Explain the type of polymorphism with code.

Ans:- The word polymorphism means having many forms. Real-life example of polymorphism, a person at the same time can have different characteristics. Polymorphism is considered as one of the important features of Object Oriented programming.

There are two types of polymorphism:-

1. Compile time polymorphism:- In this, a function is called at the time of program compilation. We call this type of polymorphism as early binding or static binding. Function overloading and operator overloading is the type of compile time polymorphism.

Ex:- #include <iostream>
using namespace std;
void test(int i) {
 cout << "The int is " << i << endl;
}
void test(double f) {
 cout << "The float is " << f << endl;
}
void test(char const *ch)
{
 cout << "The char * is " << ch << endl;
};

2. Parameterized constructors: This is the constructor with parameter. This constructor you can provide different values to data members of different objects, by passing the values as argument.

for example:

```
class Cube
{
    public:
        int side;
        Cube (int x)
        {
            side = x;
        }
        ~Cube()
        {
            int main()
            {
                Cube c1(10);
                Cube c2(20);
                Cube c3(30);
                cout << c1.side;
                cout << c2.side;
            }
        }
    }
```

Output:

10

20

30

Syntax:-

```
class_name(parameter1, parameter2, ...)
```

// constructor definition

}

for example:-

```
class Cube
```

```
{
```

```
public:
```

```
int side;
```

```
Cube()
```

```
{
```

```
side = 10;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
Cube c;
```

```
cout << c.side;
```

```
}
```

Output:-

10

Code for delete keyword:

```
include <bits/stdc++.h>
using namespace std;
int main()
{
    int *array = new int[10];
    delete [] array;
    return 0;
}
```

2. What are constructors? Why they are required? Explain different types of constructors with suitable example.

Ans: A constructor is a member function of a class which initialize objects of a class. In C++, constructor is automatically called when object create. It is special member function of class.

Constructor is a special method which is invoked automatically at the time of object creation. It is used to initialize the data members of new object generally.

There are three types of constructor which are following below:-

1. Default constructors: Default constructor is the constructor which doesn't take any argument. It has no parameter.

Code for new keyword:

using namespace std;

class car

{

String name;

int num;

public:

car(String a, int h) {

cout << "Constructor called" << endl;

this -> name = a;

this -> num = h; }

void enter ()

{

cin >> name;

cin >> num;

}

void display ()

{

cout << "name: " << name << endl;

cout << "num: " << num << endl;

}

};

int main ()

{

car *p = new car ("Honda", 2021);

p -> display ();

}

... namespace std;

Name- Piyush Pushkar

1

Assignment - 2

Choose the correct option.

1. Ans) b

2. Ans) A

3. Ans) C

4) Ans. B

5) Ans.

Short answer type question:-

1. Explain about new and delete keyword with code.

Ans:- New keyword:- The new operator is an operator which denotes a request for memory allocation on the Heap. If sufficient memory is available, new operator initializes the memory and returns the address of the newly allocated and initialized memory to the pointer variables.

Syntax:- `pointer-variable = new data-type;`

Delete keyword:- Delete is an operator that is used to destroy array and non-array (pointer) objects which are created by new expression. The delete operator has void return type.