Page No. Date

Assignment - 2

Ques 1. (d) Stack.

Ques 2. (C) Compiler Error in the "Derived *dp = new Base;"

Ques 3. (a) Inaccessible

Ques 4. (A) The no. of times destructor is called depends on no. of objects created.

Ques 5. (A) True.

Short answer type questions:

1) new : to allocate memory of any data type; the Syntax:

Pointer variable name = new data type;

this is to create new object for class.

delete : This is to deallocate the memory user has priveledge to deallocate the created pointer variable by this delete operator.

Syntax:

delete pointer variable name.

eg: for new:

```
int *pointV;
pointV = new int // allocating variable.
*pointV = 45;
```

for delete: In continuation from above.

```
delete pointV;
return 0;
```

Qu 2. Constructor: Constructor called automatically when object is declared as it initialise the object. constructor name same as it's class.

→ They are required to initialise the object of class.

→ Types of constructor:

(i) Default constructor
(ii) Copy constructor
(iii) Argument — / Parameter constructor.

eg:
```
class student {
    private:
        float marks;
    Public:
        student() {
            marks = 0;
        }
        student(x) {
            marks = x;
            cout << "marks:" << x;
        }
        student (student &obj) {
            marks = obj. marks;
        }
};
int main()
{
    student obj1, obj2;
    obj1 (); // default.
```

Page No.   Date

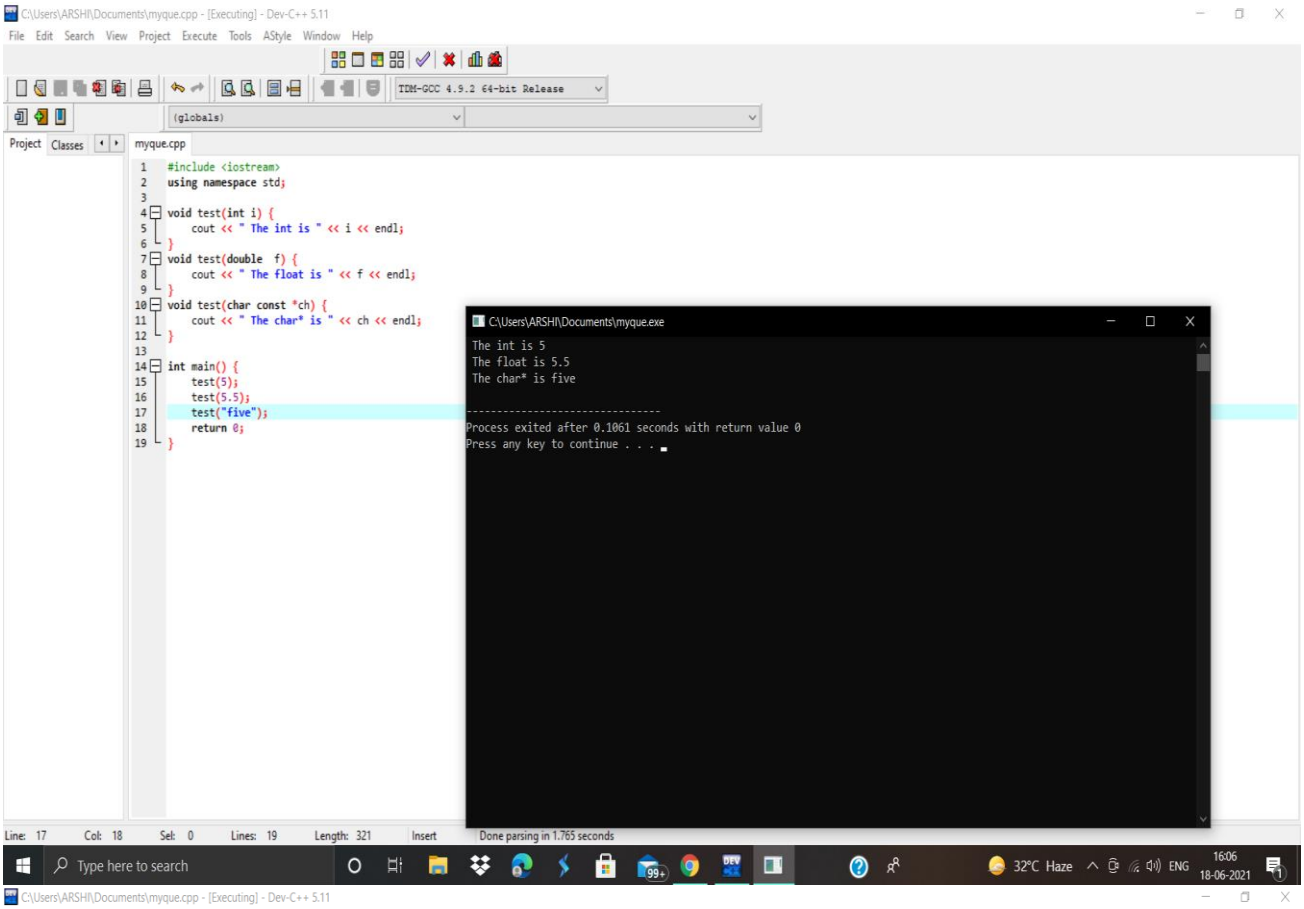
obj 2 (100); // for parameter constructor
~~Struc~~ obj 1 (obj 2); // for computer constructor.

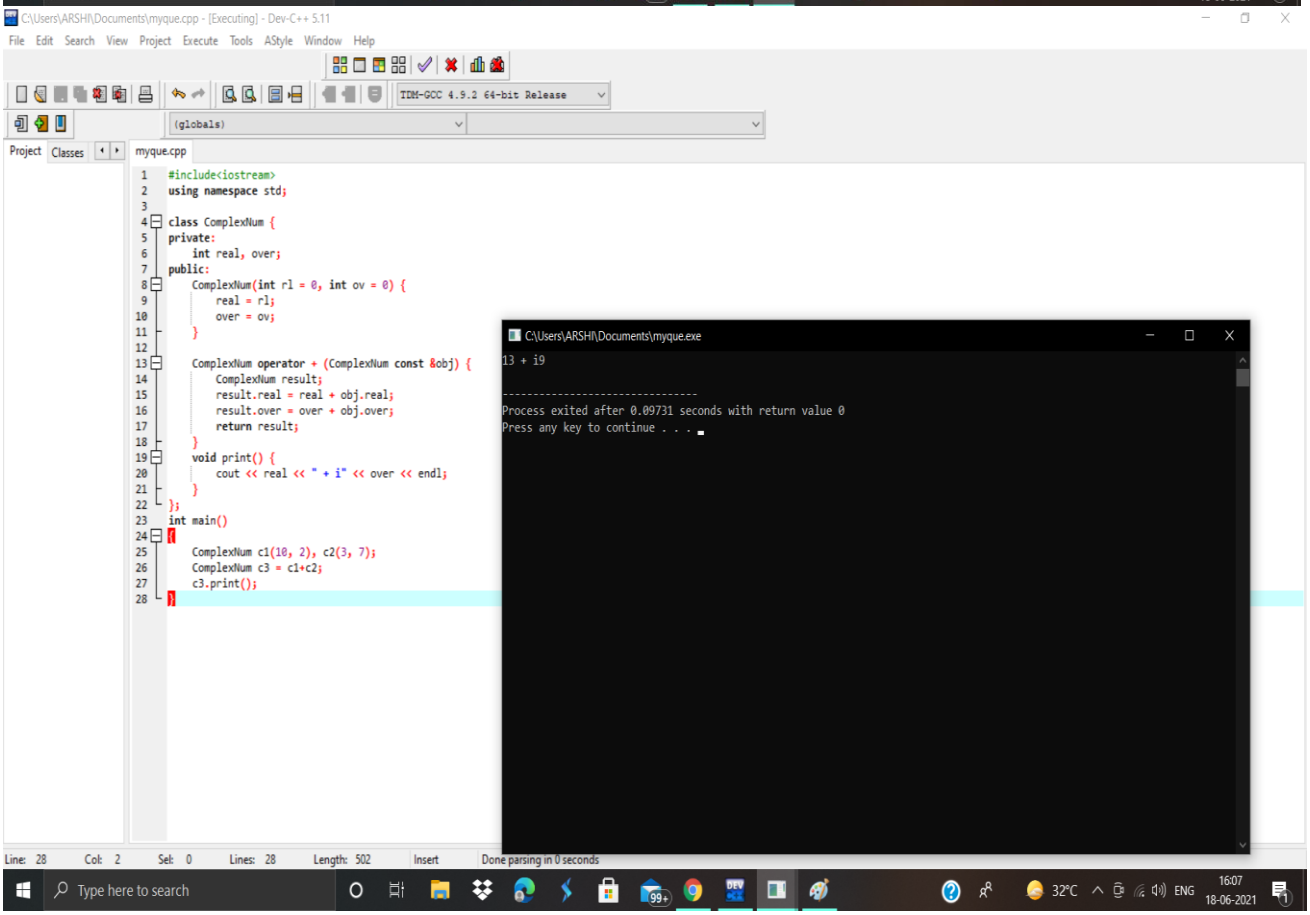
Que 3.

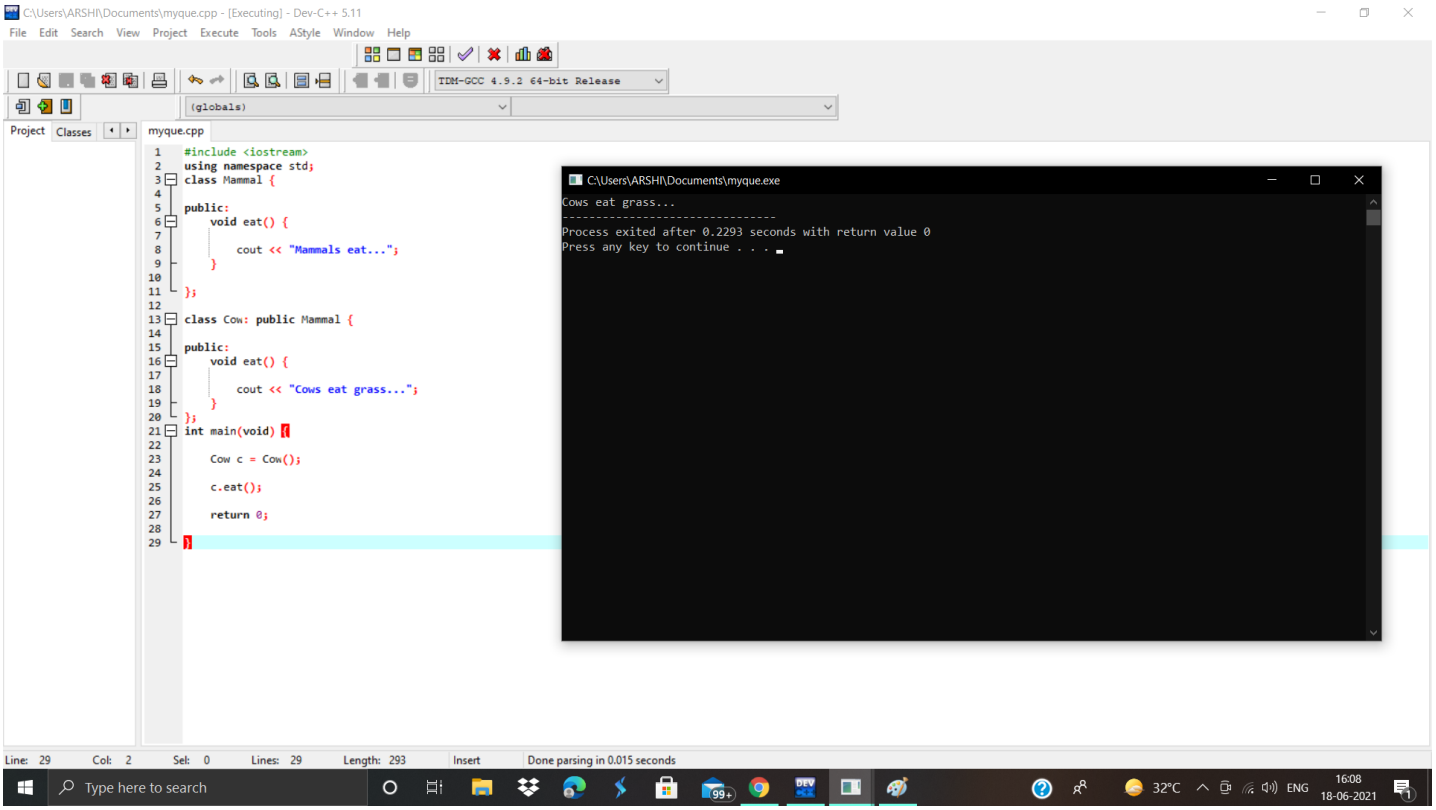
Object Oriented programming	Procedural Oriented programming.
⇒ Data hiding, class members are divided into class 3 accessible modes.	⇒ function can't hide data from their client side and user side.
⇒ wrapping up the data called class	⇒ wrapping the data is called functions.
⇒ It follow bottom-up ~~ap~~ approach	⇒ Top - down approach.
⇒ Overloading and polymorphism is available in OOPs.	⇒ Not available in procedural oriented programming.
⇒ eg: C++, Java	eg: C, Visual Basic.

Long answer type Questions.

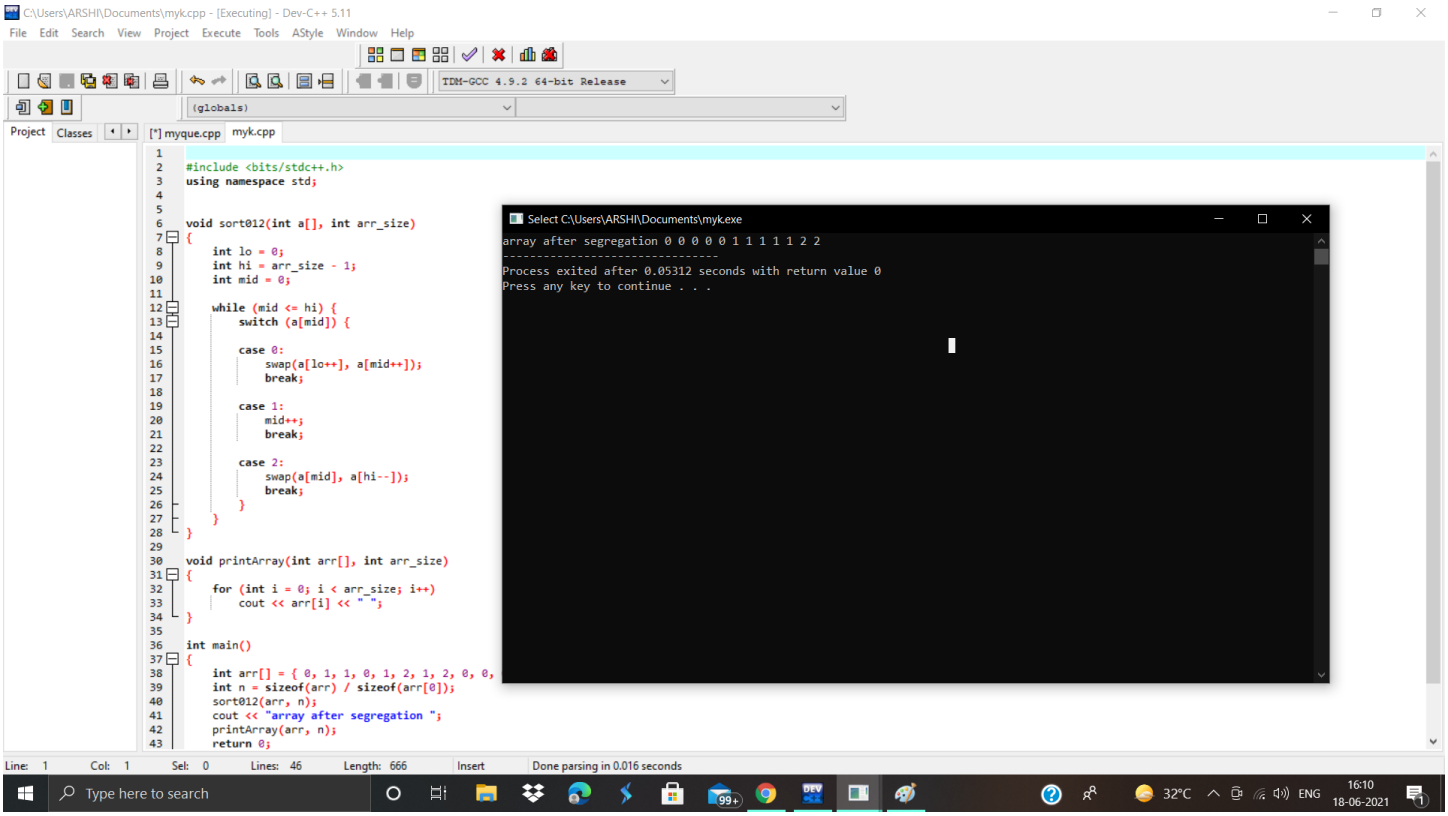
Que 1. Polymorphism: Poly- many and morphism- form i.e. many form, it means that ability of one thing to take many different forms. The one thing may be an operator or functions. eg. Overloading.


C:\Users\ARSHI\Documents\myque.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes
myque.cpp
#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;
}
int main() {
test(5);
test(5.5);
test("five");
return 0;
}
C:\Users\ARSHI\Documents\myque.exe
The int is 5
The float is 5.5
The char* is five
Process exited after 0.1061 seconds with return value 0
Press any key to continue . . .
Line: 17 Col: 18 Sel: 0 Lines: 19 Length: 321 Insert
Done parsing in 1.765 seconds
Type here to search
32°C Haze
ENG
16:06
18-06-2021


C:\Users\ARSHI\Documents\myque.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes
myque.cpp
#include<iostream>
using namespace std;
class ComplexNum {
private:
int real, over;
public:
ComplexNum(int rl = 0, int ov = 0) {
real = rl;
over = ov;
}
ComplexNum operator + (ComplexNum const &obj) {
ComplexNum result;
result.real = real + obj.real;
result.over = over + obj.over;
return result;
}
void print() {
cout << real << " + i" << over << endl;
}
};
int main()
{
ComplexNum c1(10, 2), c2(3, 7);
ComplexNum c3 = c1+c2;
c3.print();
}
C:\Users\ARSHI\Documents\myque.exe
13 + i9
Process exited after 0.09731 seconds with return value 0
Press any key to continue . . .
Line: 28 Col: 2 Sel: 0 Lines: 28 Length: 502 Insert
Done parsing in 0 seconds
Type here to search
32°C
ENG
16:07
18-06-2021


C:\Users\ARSHI\Documents\myque.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes
myque.cpp
#include <iostream>
using namespace std;
class Mammal {
public:
void eat() {
cout << "Mammals eat...";
}
};
class Cow: public Mammal {
public:
void eat() {
cout << "Cows eat grass...";
}
};
int main(void) {
Cow c = Cow();
c.eat();
return 0;
}
C:\Users\ARSHI\Documents\myque.exe
Cows eat grass...

Process exited after 0.2293 seconds with return value 0
Press any key to continue . . .
Line: 29 Col: 2 Sel: 0 Lines: 29 Length: 293 Insert Done parsing in 0.015 seconds
Type here to search
32°C ENG 16:08 18-06-2021


C:\Users\ARSHI\Documents\myk.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes
[*] myque.cpp
myk.cpp
#include <bits/stdc++.h>
using namespace std;
void sort012(int a[], int arr_size)
{
int lo = 0;
int hi = arr_size - 1;
int mid = 0;
while (mid <= hi) {
switch (a[mid]) {
case 0:
swap(a[lo++], a[mid++]);
break;
case 1:
mid++;
break;
case 2:
swap(a[mid], a[hi--]);
break;
}
}
}
void printArray(int arr[], int arr_size)
{
for (int i = 0; i < arr_size; i++)
cout << arr[i] << " ";
}
int main()
{
int arr[] = { 0, 1, 1, 0, 1, 2, 1, 2, 0, 0,
int n = sizeof(arr) / sizeof(arr[0]);
sort012(arr, n);
cout << "array after segregation ";
printArray(arr, n);
return 0;
Select C:\Users\ARSHI\Documents\myk.exe
array after segregation 0 0 0 0 0 1 1 1 1 1 2 2

Process exited after 0.05312 seconds with return value 0
Press any key to continue . . .
Line: 1 Col: 1 Sel: 0 Lines: 46 Length: 666 Insert Done parsing in 0.016 seconds
Type here to search
32°C ENG 16:10 18-06-2021

```
class Member {
Public :
    int age, Phone_no., Salary;
    char Name[10], Address[20];
  void printSalary()
    {
    Cout<<" in Salary = "<< Salary;
    }
};

class Manager : Public member {
        char specialization[30];
    void input() {
    cin>> Name>> age>> Phone_no.>> Address;
cin>>specialization;
    }
    void output() {
    cout << Name<< age << Phone_no. << Address;
    cout<< specialisation;
    }};
class Employee : public member {
        char depart-ment-[30];
    void input()
cin>>Name>> age>> Phone_no >Address>> department
    }
    void output-() {
    cout << Name<< age << phone_no << Address;
    cout << depart-ment-
    }
};
```

```
int main()
{ employee e1;
  manager m1;
  e1.input();
  m1.input();
  e1.output();
  e1.printsalary();
  m1.output();
  m1.printsalary();
}
```