Name- Sumit Kumar Giri

Course- B.Tech (CSE)

Uni. roll no.- 2000213 (CS-2nd)

Assignment- 3rd

Subject- OOPS (ACCS-16302)

Sem. - 3rd.

Section-A

1.

(i) what is the use of pure virtual function?

Ans:- pure virtual function are used to buid abstract Classes. These classes cannot be instantaitated. This means we cannot create objects of their type. this feature comes very much in handy when building interfaces such as APIs. Interfaces usually encapsulate an abstract level of generality for your code. This means you are aware that you'll used a type of object in your app but you don't yet know what its functionalitis will be.

(ii) what do you mean by Generic function?

Ans:- Generics is the idea to allow type (Integer, String,— etc and user defined type) to be a parameter to methods, classes and interfaces. for examples, classes like an array, map etc. which can be used using generics very efficiently. we can use them for any type.

(iii) Differentiate between function overloading and operator overloading?

Ans:- There are many differentiate between function overloading and operator overloading.

Function overloading	operator overloading
Function overloading allows us to Call it in multiple ways.	operator overloading allows operators to ha their extending meaning beyond its predefined operational meaning.
You Can overload the function with the Same name but with different parameters.	You Can overload (defi Custom behaviour) for operatos Such as '+', '- '()', [].
Function overloading means using a single name and giving more functionality to it.	operator overloading means adding extra functionality for a Certain operator.

(iv) Define static binding.

Ans:- In static binding, the Compiler makes the decision regarding selection of appropriate function to be Calle in response to function Call at Compile time. This is because all the address information requires to Call a function is known at Compile time. It is also known as early binding.

(V) write file

Ans:- ios::in

ios::out

ios::ate

ios::app

ios::trun

ios::nocr

ios::nore

(vi) what is

Ans:- A virtua

Space allo

instance

class usin

or parent

that ensure

destructor w

the derived

release the

(V) write file opening opening modes in c++.

Ans :- ios::in – open for input (default for ifstream)
ios::out – open for output (default fos of stream)
ios::ate – open and go to end of file.
ios::app – appends output to the end of file.
ios:: trunc – delete file contents if it exits.
ios:: nocreate – open falls if file does not exists.
ios:: noreplace – open fails if file already exits.

(vi) what is the use of virtual destructors?

Ans :- A virtual destructors is used to tree up the memory space allocated by the derived class object or instance while deleting instances of the derived class using a base class pointer object. A base or parent class destructor use the virtual keyword that ensure both base class and the derived class destructor will be called at run time, but it called the derived class first and then base class to release the space occupied by both destructors.

Section - B

Q2. Differentiate between static and dynamic binding How is dynamic binding achieved in C++? Explain with example.

Ans:- There are many differentiate between static and dynamic binding.

Static binding	Dynamic binding
The function call resolve at Compile time polymorphism (static binding). It is also known as early binding.	The function call is resolv at run time polymorphism (dynamic binding). It is a known as late binding.
It can be used with any data type.	It can be used only wit class type.
Selection of appropriate function to be invoked depends on the type and the number of arguments in the function call.	Selection of appropriate function to be invoked de on the contents of the base class pointer.
It is implemented using function overloading and operator overloading.	It is implemented using virtual function.
Its disadvantage is the lacked flexibity.	It's disadvantage is the little loss of executi speed.

Dynamic binding is achieved using virtual function Base class pointer pointer to derived Class object and a function is declared virtual in base class, then matching function is identified at run time using virtual table entry.

```
# include<iostream.h>
# include<conio.h>
   Class A
   {
     public :
     virtual void display()
     {
      Cout<<"Base";
      }
      };
     Class B : Public A
      {
       public :
     void display()
     {
      Cout<<"Derived";
       }
      }; int main()
       {
         B b;
      A*a =&b;
      a->display();
       return 0;
       }
```

Q3. Discuss about various file opening modes in C++. Write a program to write the content into a file and then read the same using file handing in C++.

Opening mode	Effect.
ios :: in ios :: out	open for input (for default ifstream). open for output (for default ofstream).
ios : : ate	open and go to a end of file.
ios : : app	Appends outputs to end of file.
ios :: trunc	Delete file Contents if it is exit.
ios : : no create	open fails if a file does not exit.
ios : : no replace	open fails is already exist.
ios : : binary	opens as binary file.

Program to write and read Content :

```
# include<iostream.h>
# include< fstream.h>
# include< conio.h>
 int main()
 {
  Clrscr();
 int roll no., marks;
Char name[20];
ofstream fout ;
{
 fout.open ("New.txt");
 fout<<" Hi this is me";
}
  {
   ifstream fin;
   fin.open ("New.txt");
   Char.ch;
   ch= fin.get ();
   while (! fin.eof());
  {
   Cout<<ch;
   ch= fin.get();
  }}
    getch();
   }
```

OUTPUT :-

```
Hi, this is me.
```

Q4. write a program to find the Cube of a number using function template.

Ans:

```
#include <iostream.h>
 # include< conio.h>
  us
   template <class T>
   T add (T a)
  {
   return(axaxa);
  }
   int main()
   {
    int c,a;
    float b;
    s;
   Cout << "\n Enter the value type : \n1)
                 Integer \n2)
    float \n select [1,2] = ";
    cin >> c;
   Switch (c)
    {
     Class 1 :
     {
     Cout<< "Enter one Integer Number: ";
     Cin>> a;
     Cout << add << int> (a);
      getch();
    go to s;
    }
```

```
case · 2:
 {
  Cout << " Enter · one float number: ";
   Cin >> b;
   Cout << add < float >(b);
     getch();
     go to S;
    }
     default
    Cout << " Invalid " << endl;
   go to S;
    return 0;
   }
```

OUTPUT

```
Enter the value type:
    1> Integer
    2> float
     Select [1,2]=1
    Enter one Integer Number: 5
       125
```