

OOPS Assignment 2

- 1) Assume that an integer and a pointer each takes 4 bytes. Also, Assume that there is no alignment in objects. Predict the output of the following program.

```
#include <iostream>
using namespace std;
class Test
{
    static int x;
    int * ptr;
    int y;
};
int main()
{
    Test t;
    cout << size of (t) << " ";
    cout << size of (Test *) << " ";
}
```

The output will be
16 8

- 2) Identify the error in the following program:-

```
#include <iostream>
using namespace std;
class info
{
    char * name;
    int number;
public:
    void getinfo()
    {
        cout << "info::getinfo";
        getname();
    }
    void getname()
    {
        cout << "info::getname";
    }
};
class name; public info
{
    char * name;
```

public:

```
void getname()
{
    cout << "info :: getname ";
}
};

void main()
{
    info * p;
    name n;
    p = n;
    p->getinfo();
    return 0;
}
```

- 1) Syntax errors:-> The 'u' in the using namespace std should be lowercase
- > The 'c' in class info should be lowercase
 - > The 'c' in cout should be lowercase
 - > The 'v' in void main should be lowercase
- 2) Logical errors:-
- > Inheritance and polymorphism:- The program attempts to use inheritance and polymorphism, but there is a mistake in how this is implemented
 - > Pointer assignment:- 'p = n'; is incorrect because p is a pointer to info but n is an object of type name.

Q3) In C++, a pure function is a virtual function that is declared in the base class without definition, and ends with "= 0" which is declared. A pure virtual function is also known as abstract function.

In C++ the 'this' pointer is a special pointer that points to the object for which a member function is being called. It is passed implicitly as a hidden argument to all non-static member function calls. The 'this' pointer points to the object itself.

•) At compile time polymorphism, the compiler uses static or early binding to determine the correct method to be called based on the type of pointer or reference being used.

Compiler-time polymorphism is used achieved by ~~on~~ function overloading and operator overloading where the compiler determines which function or operator implementation to use based on the function signature or operator applied.

•> Runtime polymorphism requires the use of virtual functions in the base class. When a derived class overrides ~~the~~ a virtual function, the method call is determined dynamically based on the actual object type at runtime.