1601-14-733-091 CSE 2 BE Y4 II nd sem

Laboratory Record

of

Sheet No.

Experiment No

Date

DIFFERENCE BETWEEN C & C++

C	C++
(1) Procedure Oriented programming language.	Object Oriented Programming Language.
(2) Top-down approach is followed.	bottom-up approach is followed.
(3) There is no-protection and safety of data.	Protection of data is ensured by using objects created by classes.
(4) Uses structures	Both structures and classes can be used.
(5) Encapsulation, Data hiding, polymorphism, abstraction, function over-loading etc are some features present in C++ by not present in C.	Encapsulation Data hiding polymorphism. abstraction function overloading reusable code

Roll No.

CBIT

C	C++
(6) Emphasis is on ~~data~~ procedure rather than data.	Emphasis is on data rather than procedure.
(7) stdout (output stream) stdin (input stream)	cin (input object) cout (output object)
(8) Programs divided into functions.	Programs divided into objects
(9) Data moves openly around the system from function to function.	Data is hidden and cannot be accessed by external functions.

Laboratory Record 1601-14-733-091 CSE 2 BE 1/4

of

Sheet No.

Experiment No.

Date

DIFFERENCE BETWEEN CLASS AND A STRUCTURE

CLASS	STRUCTURE
(1) Related to Object oriented programming	Related to Procedure Oriented Programming
(2) Classes are blue prints to create objects. Objects contain data as well as functions needed to manipulate the data. (Typedef not required.)	Structures are ~~hete~~ user defined heterogeneous data types. It is a collection of various primitive data types like int, char, float etc. (Typedef is required)
(3) Each member data ~~type~~ of the object cannot be ~~asss~~ accessed directly.	Each member can be accessed directly.
(4) The data of the object created by class is hidden, it can be modified or used by the interior functions of the object only. This ensures ~~ensures~~ the data safety.	each member of the structure can be ~~case~~ accessed or modified by outside functions also. Hence the original data is at risk.

Roll No. CBIT

Memory is allocated for both data and functions in class , for structures memory is allocated for the data only….

Ex:-

Class

```
class student
{
  char name[100];
  int rank, roll;
  public:
  void getdata()
  {
   cin>>name>>roll>>rank;
  }
  void display()
  {
  cout<<name<<"\n"<<roll<<rank;
  }
};
```

Object declaration :-
(in main())

Student stu;

upon declaration of stu, the variables name, rank, roll
(name: string, rank: int, roll: int)
can be accessed by the functions getdata() and display() only.
stu.getdata(); stu.display(); in main
hence ensures safety.

Structure

```
typedef struct
{
  char name[100];
  int rank, roll;
} Student;
```

User defined variable declaration :-
(in main())

Student stu;

in the structure variable stu, the variables name, rank, roll can be accessed by all the functions in the program. (hence no data safety).