Exception Handling
- try, catch, throw

2 hrs

```
divsion(int n,int d){
        throw 1.2;
}

try{
division();
multiplication();
}
catch(double){

}
```

STL -> Standard Template Library

- It is a library that consists of all the template functions and template classes
- It has 4 components

1. Container ->
2. Algorithm ->
3. Functors (Function Objects)
4. Iterators

Container

- It is a data structure template classes that are used to store the data
- Types of Containers

1. Sequence Containers -> vector
2. Associative Containers -> set, map
3. Unordered Associative Containers -> set,map
4. Sequence Adapter Containers -> stack,queue

```
vector<Employee *> v1;
itr = v1.begin();
```

vector<Employee>
vector::iterator<Employee>
vector<Employee*>::iterator;
non of the above

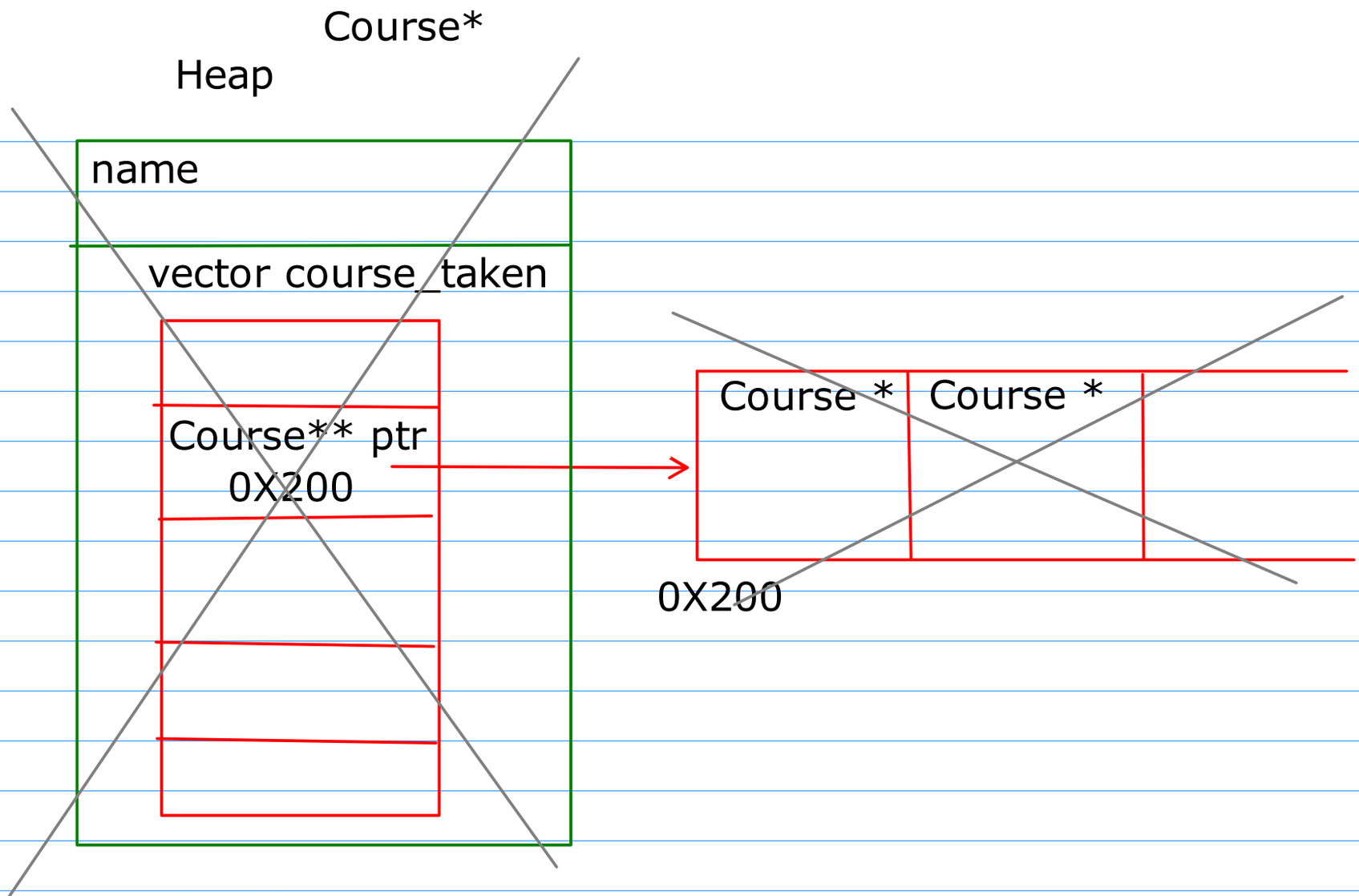
```
itertor itr{
int *ptr;
iterator(double *ptr)
}
```

```
class Course{
int cid;
string course_name;
double fees;
.
.
.
}
```

```
class Student{
int rollno;
string name;
vector<Courses *> course_taken;

}
```

```
vector<Student *> student_list;
vector<Courses *> courses_list;
```



delete student_list[i]

FILE IO
Copy Ctor (Deep and Shallow)
Operator Overloading
nested and local class
singleton design pattern
factory design pattern
Smart pointers