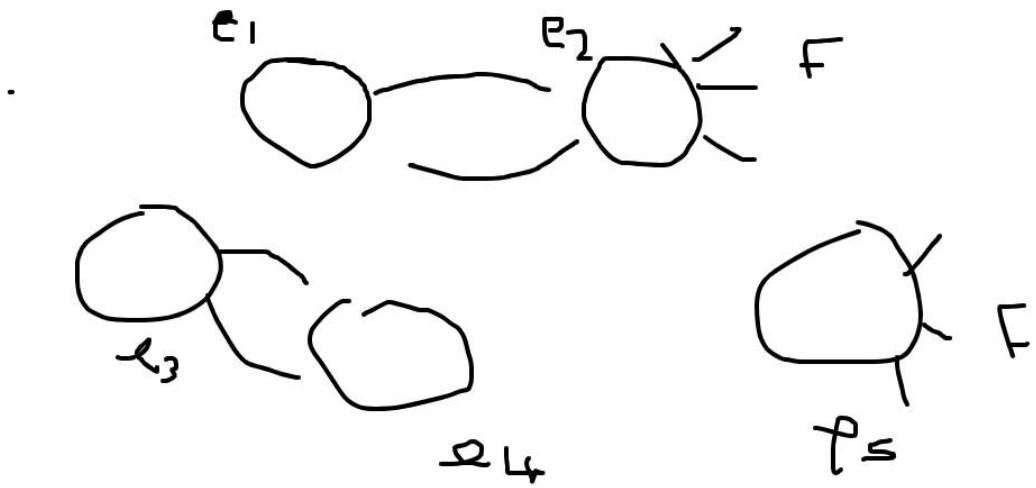


Requirment--->Define pb in understandable way
"get grades of students"

Design ----->transfer problem into documents ERD
,ClassDiagram,Usecases,flowcharts ,UI screens.

Implementation ---->code File Source files based technology based
team role

testing--->unit testing, QA team

Deployment---->publishing

Maintenance---->Versioning (update)

main.cpp source file--->(header files (#)),function main (code)

Build(1-preprocessor->expanding header files used in this source file create a tempfile in

memory *.i)
2-compiler-->check source file syntax semantics free of errors(list of errors or warnings)-
-
>succeed *.o(machine understandable code file-intermediate language)
3-Linker--->connect link all object files that my project need to run   (main.o+libraries)

file1.cpp---->file1.o
void greet(){ cout<<hi;}
linker ---------->main.o+file1.o+libraries-->executable file *.exe
main.cpp---->main.o
int main(){   greet();return 0;}

static linking---> copying *.o files into main.o file   int main(){cout<<hi;return 0;}
exe file (large size) independent
dynamic linking--->connect files in execution.

loader-->exe machine

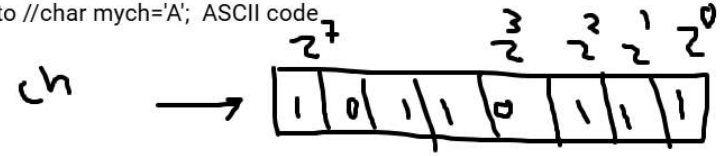
Variables & Memory

Memory Layout Concept
When a C++ program runs, its memory is divided into:
• Code segment – machine instructions.
• Stack – stores local variables and function calls (LIFO).
• Heap – dynamic memory (new, delete, or smart pointers).
• Global/Static area – global variables and static objects.

Variables: location reserved in memory to store specific or variable values .

int -->4 bytes sizeof(int)--->how many bytes in memory for this types
int x; declare variable named x
int x=5; initialization ---- auto x=5; compiler identify type by value
float --->8 bytes float z=5.2f;
double -->16 byte double n=667.88;

char mych; -->1 byte from memory 8 bits 0-->255 as a decimal value
auto //char mych='A'; ASCII code



0-->32 decimal non printable keys chars
33-->127 decimal printable keys a(97)---->z lower case A(65)-->Z uppercase

0--
>9
some of extended keys127 -->255

string is a text (sequence of chars) --->char[ ]
std::string name; std::string//auto name="sheryyy";

bool flag=(true /false)

input output operations (cin,cout) objects not functions

```
int x,y;
cin>>x>>y;
cin>>x;
cin>>y;
```

hint-->terminator is char '\0'