***SOFTWARE DEVELOPMENT LAB- II***

***WEEK 6A***

Q1.

#include <iostream>

using namespace std;

class student

{

private:

char \*name\_s;

public:

student()

{

cout<<"Student constructor\n";

}

~student()

{

cout<<"Student destructor\n";

}

void sname()

{

cout << "Enter the name of the student\n";

cin >> name\_s;

}

};

class department

{

char \*name\_d;

public:

department()

{

cout<<"Department constructor\n";

}

~department()

{

cout<<"Department destructor\n";

}

void dname(){

cout << "Enter the name of the department\n";

cin >> name\_d;

}

};

class course

{

student std\_p;

department dept\_p;

char \*course\_name;

static unsigned int index;

static course courselist\_p;

};

int main()

{

student s;

department d;

s.sname();

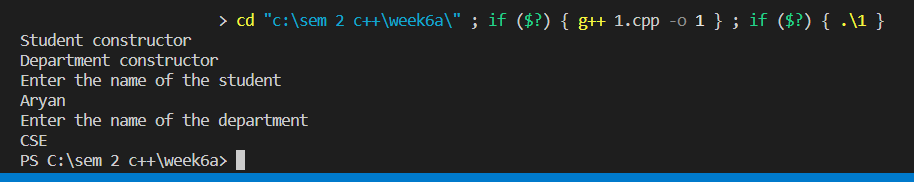
d.dname();

s.~student();

d.~department();

return 0;

}



Q2.

#include<iostream>

using namespace std;

class Employee

{

char \* myname\_p;

public:

void dispaly();

Employee()

{

cout<<"Employee joined\n";

}

~Employee()

{

cout<<"Employee fired\n";

}

};

class Company

{

public:

char \* name\_p;

Employee \* myemp\_p;

public:

Company()

{

cout<<"company started \n";

myemp\_p=new Employee;

}

~Company()

{

cout<<"Company closed\n";

}

};

void Employee :: dispaly()

{

cout<<"Enter the name of the employee\n";

cin>>myname\_p;

}

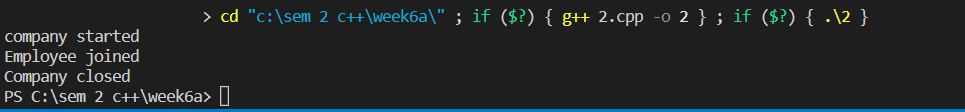
int main()

{

Company c;

return 0;

}



Q3.

#include<iostream>

using namespace std;

class Room

{

char \* name\_p;

public:

Room(){

cout<<"Room created\n";

}

~Room(){

cout<<"Room destroyed\n";

}

static void initlist\_v(){}

static void createRoom\_v(){}

void display(){}

};

class House

{

char \* name\_p;

Room rommlist\_p;

public:

House(){

cout<<"House crreated\n";

}

~House(){

cout<<"House destroyed\n";

}

void display(){}

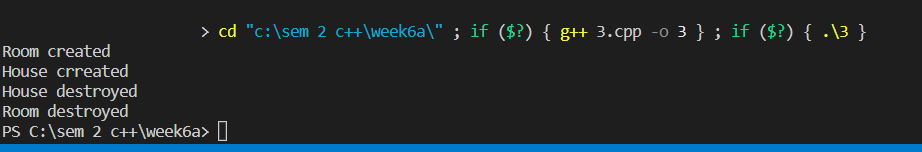
};

int main(void)

{

House h;

}



Q4

#include<iostream>

using namespace std;

class wheel{};

class person{};

class vechile

{

string wheelbase;

int maxpower;

wheel w;

person \*p;

protected:

int seat;

public:

vechile()

{

cout<<"Vechile created\n";

}

void pressAccelerator(){}

void pressbrake(){}

void turnwheel(){}

};

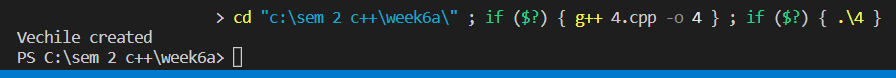
int main()

{

vechile v;

return 0;

}



Q5.

#include<iostream>

using namespace std;

class wheel{

public :

wheel()

{

cout<<"Wheel installed\n";

}

~wheel()

{

cout<<"Wheel destroyed\n";

}

};

class person{};

class tachograph

{

public:

tachograph()

{

cout<<"Tachogarph installed\n";

}

~tachograph()

{

cout<<"Tachogarph busted\n";

}

};

class lorry

{

tachograph \* t;

public:

lorry(){

t=new tachograph;

cout<<"Lorry manufactured\n";

}

~lorry()

{

cout<<"Lorry destroyed\n";

}

};

class vechile

{

string wheelbase;

int maxpower;

wheel \* w;

person \*p;

protected:

int seat;

public:

vechile()

{

cout<<"Vechile manufactured\n";

w= new wheel;

}

~vechile()

{

cout<<"Vechile crashed\n";

}

void pressAccelerator(){}

void pressbrake(){}

void turnwheel(){}

};

int main()

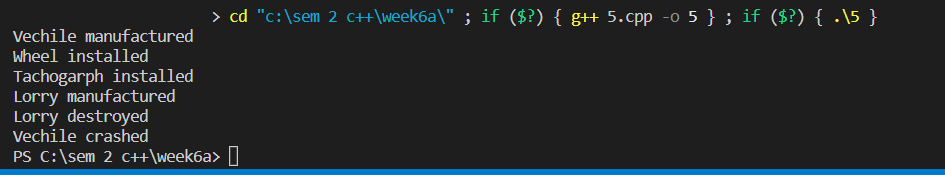
{

vechile v;

lorry l;

return 0;

}



Q6.

#include<iostream>

using namespace std;

class vechile

{

string wheelbase;

int maxpower;

protected:

int seat;

public:

vechile()

{

cout<<"Vechile constructor\n";

}

~vechile()

{

cout<<"Vechile destroyed\n";

}

void pressAccelerator(){}

void pressbrake(){}

void turnwheel(){}

};

class lorry : public vechile

{

public:

lorry()

{

cout<<"Lorry constructor\n";

}

~lorry()

{

cout<<"Lorry destructor\n";

}

};

class car : public vechile{

public:

car()

{

cout<<"Car constructor\n";

}

~car()

{

cout<<"Car destructor\n";

}

};

int main()

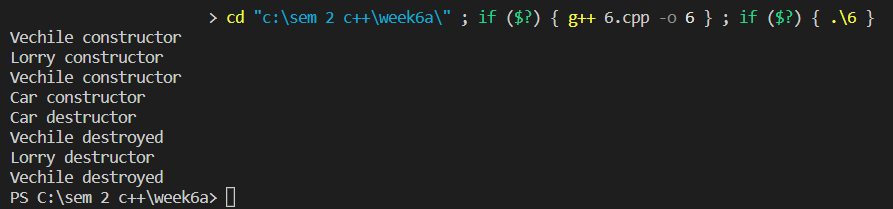
{

lorry l;

car c;

return 0;

}



Q7.

#include<iostream>

using namespace std;

class person{

public :

person()

{

cout<<"Person booked car\n";

}

~person()

{

cout<<"Person dead\n";

}

};

class vechile

{

string wheelbase;

int maxpower;

person \*p;

protected:

int seat;

public:

vechile()

{

cout<<"Vechile created\n";

p=new person;

}

~vechile()

{

cout<<"Vechile crashed\n";

}

void pressAccelerator(){}

void pressbrake(){}

void turnwheel(){}

};

int main()

{

vechile v;

}

