**\_\_\_\_\_**

Q1 :

1. Declare a **Book** class consist of the book title, number of pages, publisher, author .Those variables are private.
2. Add functions to the class members in order to set and get the variables.
3. Add a print function to print the book information all information (book title, number of pages, publisher, author)
4. Add two constructors:
   1. Default constructor (do nothing).
   2. Constructor to set the **book pages** with a given value.
5. Add a destructors to print a destroy massage.

**Q2 : complete the program. You may need to make some changes in the code:**

class Dog {

protected:

int age;

\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

};

int main(){

//declare a dog called woofy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// set Woofy’s age to 5

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

return 0;

}

**3- Find ERROREs in this programme and fix them**

#include <iostream>

#include <string>

using namespace std;

class Celcius{

private:

float c;

void readCelcius(){

cout<<"Enter temperature in Celcius: ";

cin>>c;

}

public:

//constructor

Celcius(){ c= 0};

Celcius( float c1) { c= c1;}

// destructor

float ¬ Celcius(int x);

float toFarenheit(){return((9.0/5.0)\*c+32);}

};

int main(){

Celcius c1;

c1.Celcius();

Celcius c2();

Celcius c2(3.4);

c1.readCelcius();

cout<<c1.toFarenheit()<<" F"<<endl;

f1.readFarenheit();

cout<<f1.toCelcius()<<" C";

return 0;

}

4. trace the following program ,show the output

#include <iostream>

#include <string>

using namespace std ;

//==================================================

class CreateAndDestroy {

public:

CreateAndDestroy(int objectNumber, string message )// constructor

~CreateAndDestroy(); // destructor

private:

int objectID;

string message;

}; // end class CreateAndDestroy

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// constructor

CreateAndDestroy::CreateAndDestroy( int objectNumber, string message )

{

objectID = objectNumber;

message = message;

cout << "Object " << objectID << " constructor runs "

<< message << endl;

} // end CreateAndDestroy constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// destructor

CreateAndDestroy::~CreateAndDestroy()

{

cout << "Object " << objectID << " destructor runs "

<< message << endl;

} // end ~CreateAndDestroy destructor

//==================================================

void create( void ); // prototype

// global object

CreateAndDestroy first( 1, "(global before main)" );

int main()

{

cout << "\nMAIN FUNCTION: EXECUTION BEGINS" << endl;

CreateAndDestroy second( 2, "(local in main)" );

create(); // call function to create objects

cout << "\nMAIN FUNCTION: EXECUTION RESUMES" << endl;

CreateAndDestroy third( 3, "(local in main)" );

cout << "\nMAIN FUNCTION: EXECUTION ENDS" << endl;

return 0;

} // end main

//===================================================

// function to create objects

void create( void )

{

cout << "\nCREATE FUNCTION: EXECUTION BEGINS" << endl;

CreateAndDestroy fourth( 4, "(local in create Function )" );

CreateAndDestroy fifth(

5, "(local in create Function)" );

cout << "\nCREATE FUNCTION: EXECUTION ENDS" << endl;

} // end function create