Virtual Work : OOP Concepts.

Write a program to find the factorial of a number method declared inside the class.

1:

--------------------------------------[Source Code]-------------------------------------

#include <iostream>  
using namespace std;  
class factorial {  
public:  
 void fact\_method() {  
 int i{0}, x{0}, \*P;  
 float x\_factorial{1};  
 cout << "\nEnter a positive integer: " << endl;  
 cin >> x;  
 P = &x;  
 // loop condition is not fullfilled until i has counted up(loop has  
 // repeated) to the value of '\*P'  
 for (i = 1; i <= \*P; i++) {  
 x\_factorial \*= i;  
 }  
 cout << "Factorial: [" << x\_factorial << "]" << endl;  
 }  
};  
int main() {  
 //initialize class  
 factorial x\_fact{};  
 bool exit{false};  
 string user\_exit;  
 do {  
 x\_fact.fact\_method();  
 // allows for user to find factorial of multiple numbers  
 cout << "Continue?(y/n): [ ]\b\b";  
 cin >> user\_exit;  
 if (user\_exit != "y") {  
 exit = true;  
 }  
 } while (exit != true);  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

Enter a positive integer:

5

Factorial: [120]

Continue?(y/n): [y]

Enter a positive integer:

9

Factorial: [362880]

Continue?(y/n): [y]

Enter a positive integer:

7

Factorial: [5040]

Continue?(y/n): [y]

Enter a positive integer:

11

Factorial: [3.99168e+07]

Continue?(y/n): [n]

Process finished with exit code 0

###########################################################

###########################################################

Write a program to find the factorial of a number using methods outside the class.

2:

--------------------------------------[Source Code]-------------------------------------

#include <iostream>  
using namespace std;  
class factorial {  
public:  
 //Decleration of function  
 void fact\_method();  
};  
//definition of function  
void factorial::fact\_method() {  
 int i{0}, x{0}, \*P;  
 float x\_factorial{1};  
 cout << "\nEnter a positive integer: " << endl;  
 cin >> x;  
 P = &x;  
 // loop condition is not fullfilled until i has counted up(loop has  
 // repeated) to the value of '\*P'  
 for (i = 1; i <= \*P; i++) {  
 x\_factorial \*= i;  
 }  
 cout << "Factorial: [" << x\_factorial << "]" << endl;  
}  
  
int main() {  
 //initialize class  
 factorial x\_fact{};  
 bool exit{false};  
 string user\_exit;  
 do {  
 x\_fact.fact\_method();  
 // allows for user to find factorial of multiple numbers  
 cout << "Continue?(y/n): [ ]\b\b";  
 cin >> user\_exit;  
 if (user\_exit != "y") {  
 exit = true;  
 }  
 } while (exit != true);  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

Enter a positive integer:

5

Factorial: [120]

Continue?(y/n): [y]

Enter a positive integer:

9

Factorial: [362880]

Continue?(y/n): [y]

Enter a positive integer:

7

Factorial: [5040]

Continue?(y/n): [y]

Enter a positive integer:

11

Factorial: [3.99168e+07]

Continue?(y/n): [n]

Process finished with exit code 0

###########################################################

###########################################################

Write a C++ program to find the square of an integer number and square of a double number and count number of times a function is called.

3:

--------------------------------------[Source Code]-------------------------------------

#include <iostream>  
using namespace std;  
static int cnt{0};//global varriable to keep count of times both function have been called  
int sqr\_int(int x) {  
 int prod\_x = (x \* x);  
 cnt++;  
 cout << "Square of int: " << prod\_x << endl;  
 return 1;  
}  
double sqr\_double(double y) {  
 double prod\_y = (y \* y);  
 cnt++;  
 cout << "Square of int: " << prod\_y << endl;  
 return 1;  
}  
  
int main() {  
 int x;  
 double y;  
 char user\_exit;  
 do {  
 cout << "\nEnter int: ";  
 cin >> x;  
 cout << "Enter double: ";  
 cin >> y;  
 sqr\_int(x);  
 sqr\_double(y);  
 cout << "Continue?(y/n): [ ]\b\b";  
 cin >> user\_exit;  
 } while (user\_exit == 'y');  
 cout << "\nTimes Called: " << cnt << endl;  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

Enter int: 6

Enter double: 7.8

Square of int: 36

Square of int: 60.84

Continue?(y/n): [y]

Enter int: 12

Enter double: 17.982230

Square of int: 144

Square of int: 323.361

Continue?(y/n): [y]

Enter int: 99

Enter double: 44.0454

Square of int: 9801

Square of int: 1940

Continue?(y/n): [n]

Times Called: 6

Process finished with exit code 0

###########################################################

###########################################################

Hint:

You may have one class square and many methods like getting input, calculation of square, displaying the output etc.,

Use static variable to count the all the methods you called in the main function.

Write a C++ program to display time in HH: MM: SS using class.

Hint: 02 hours:34 minutes:26 seconds.

4:

--------------------------------------[Source Code]-------------------------------------

-------------------------------------[OUTPUT]-------------------------------------

###########################################################

###########################################################

Write a program to create student class, read and print N (at least 4 inputs) student’s details.

Hint: Students details may be name, ID, Discipline, Year of study, age etc.,

5:

--------------------------------------[Source Code]-------------------------------------

#include <iostream>

using namespace std;

class student {

private:

string name;

string major;

int uin{};

int age{};

public:

//set to auto to allow for both string and Major values.

auto set\_user\_inf(string NAME, string MAJOR, int AGE, int UIN);

//using getter/setter format

void get\_user\_inf() {

//Clearly indicate a new listing

cout << std::string(25, '#') << endl;

cout << "Name: ";

cin >> name;

cout << "Current Major: ";

cin >> major;

cout << "Age: ";

cin >> age;

cout << "UIN: ";

cin >> uin;

}

void dispay\_user\_inf() {

cout << "\n" << std::string(25, '#') << "\nName: " << name << "\nMajor: " << major << "\nAge: " << age

<< "\nStudent UIN: " << uin << endl;

}

};

auto student::set\_user\_inf(string NAME, string MAJOR, int AGE, int UIN) {

this->uin = UIN;

this->age = AGE;

this->name = std::move(NAME);

this->major = std::move(MAJOR);

}

int main() {

student s{};

//input from user

s.get\_user\_inf();

//or

s.dispay\_user\_inf();

//input from main function

s.set\_user\_inf("Alex", "SE", 90, 987654321);

cout << std::string(25, '#') << "\n[Manual input below]" << endl;

s.dispay\_user\_inf();

return 0;

} ---------------------[User Input] ---------------------

Name: Alexander

Current Major: S.E

Age: 20

UIN: 123456789

-------------------------------------[OUTPUT]-------------------------------------

#########################

Name: Alexander

Major: S.E

Age: 20

Student UIN: 123456789

#########################

[Manual input below]

#########################

Name: Alex

Major: SE

Age: 90

Student UIN: 987654321

#########################

###########################################################

###########################################################