Experiment 3.a

Date:05-September-2018

**Aim:** To write a program to access elements of an array using pointer in c++.

**Algorithm:**

1: Start.

2: Declare a class as array with variable a[5] and \*ptr .

3: Declare a function as get\_data() to get values of array a[5] .

4: Declare a function as put\_data() to display values of array a[5].

5: Declare and define Main function.

6: Create an Object of array class.

7: Call get\_data() function using object.

8: Call put\_data() function using object.

9: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

class array

{ private: int a[5] , \*ptr ;

public: void get\_data();

void put\_data(); };

void array::get\_data()

{ cout<<"Enter 5 elements of array"<<endl;

for(int i =0 ; i<5 ; i++)

{ cin>>a[i]; } }

void array::put\_data()

{ ptr = a ;

cout<<"Entered Elemets are"<<endl;

for(int i=0 ; i<5 ; i++)

{ cout<<\*ptr<<endl;

ptr++; } }

int main()

{ array obj ;

obj.get\_data() ;

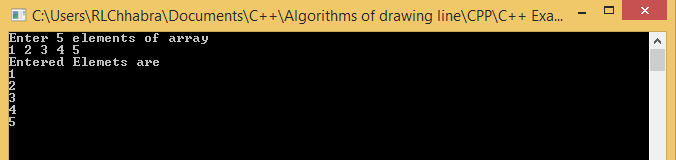
obj.put\_data();

getch() ;

return 0 ; }

**Input Given:** a[5] = { 1,2,3,4,5}

**Output:**



Experiment 3.b

Date:05-September-2018

**Aim:** To write a program to swap three variable in cyclic order in c++.

**Algorithm:**

1: Start.

2: Declare a class as Cylic\_swap with variables a , b and c.

3: Define a function as get\_a\_b\_c() to get values of a , b and c.

4: Define a function as Cyclic\_swp() to swap and display.

5: Define Main function.

6: Create object of class.

7: Call get\_a\_b\_c() function using object.

8: Call Cyclic\_swp() function.

9: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

class Cylic\_swap

{ private: int a , b , c ;

public: void get\_a\_b\_c()

{ cout<<"Enter values of a b and c"<<endl;

cin>>a>>b>>c; }

void Cylic\_swp()

{ int temp ;

temp = c ;

c = b ;

b = a ;

a = temp ;

cout<<" After swaping "<<endl;

cout<<"a = "<<a<<"\nb = "<<b<<"\nc = "<<c<<endl; } };

int main()

{ Cylic\_swap obj ;

obj.get\_a\_b\_c() ;

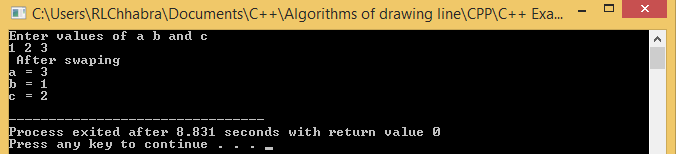
obj.Cylic\_swp();

getch();

return 0 ; }

**Input Given:** a = 1 , b = 2 , c = 3 ;

**Output:**

****

Experiment 3.c

Date:05-September-2018

**Aim:** To write a program to reverse a string using pointer in c++.

**Algorithm:**

1: Start.

2: Declare a class as reverse with string variable str and char variable a , b.

3: Define a member function as get\_string() to take input a string.

4: Define a member function as revers() to reverse string.

5: Define Main() function.

6: Create object of class.

7: Call get\_string() function using object.

8: Call revers() function using object.

9: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std;

class reverse

{ private: string str ; char a , b ;

public: void get\_string()

{ cout<<"Enter any string"<<endl;

getline(cin,str); }

string revers()

{ char \*p = & a , \*q = & b ;

int l ; l = str.size();

for(int i=0 ; i<l/2 ; i++)

{ \*p = str[i]; \*q = str[l-1-i]; str[i] = \*q ; str[l-1-i] = \*p ; }

return str ; } };

int main()

{ reverse obj ;

obj.get\_string();

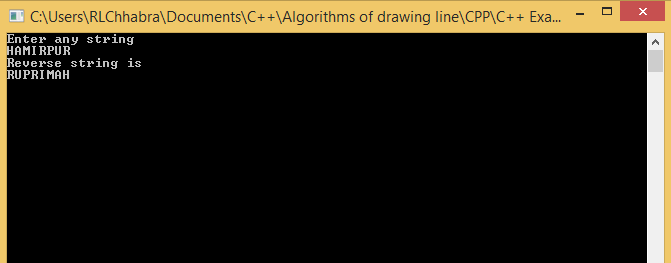
cout<<"Reverse string is "<<endl;

cout<< obj.revers();

getch(); return 0 ; }

**Input Given:** str = “ HAMIRPUR “

**Output:**

****

Experiment 3.d

Date: 05-September-2018

**Aim:** To write a program to print different size of pointer in c++.

**Algorithm:**

1: Start.

2: Declare pointer variables \*p1, \*p2 , \*p3 , \*p4 , \*p5.

Int \*p1 , char \*p2 , double \*p3 , long int \*p4 , float \*p5

3: Print different pointer size using cout.

4: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

int main()

{

int \*p1 ;

char \*p2 ;

double \*p3 ;

long int \*p4 ;

float \*p5 ;

cout<<" Size of int pointer "<<sizeof(\*p1)<<endl;

cout<<" Size of float pointer "<<sizeof(\*p5)<<endl;

cout<<" Size of Char pointer "<<sizeof(\*p2)<<endl;

cout<<" Size of double pointer "<<sizeof(\*p3)<<endl;

cout<< " size of long int pointer "<<sizeof(\*p4)<<endl;

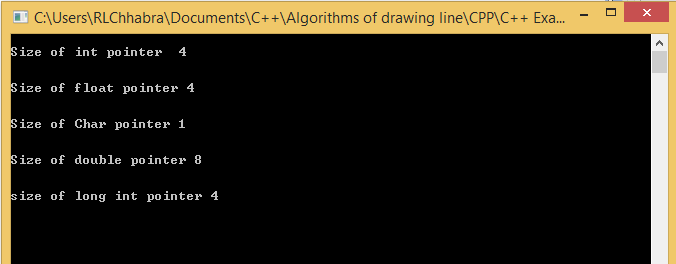
getch();

return 0 ;

}

**Input Given:** No input given.

**Output:**



Experiment 4.a

Date: 19-September-2018

**Aim:** To write a program in c++ to find factorial of a number using copy constructor.

**Algorithm:**

1: Start.

2: Declare a class as factorial with variable n and fact.

3: Declare and define constructor.

4: Declare and define copy constructor.

5: Define member function as calculation().

6: Define member function display().

7: Define main() function.

8: Create object1 of class.

9: Call calculation() function using class object.

10: Call display() function using class object.

11: Create new object2 and assign the values of object1.

12: Call calculation() function using class object2.

13: Call display() function using class object2.

14: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

class factorial

{ int n , fact = 1 ;

public: factorial(int x )

{ n = x ; fact = 1 ; }

factorial(factorial &obj)

{ n = obj.n; fact = 1 ; }

void calculation()

{ for(int i= 1 ; i<=n ; i++)

{ fact \*=i ; } }

void display()

{ cout<<"Factorial of number is = "<<fact<<endl; }};

int main()

{ factorial obj(5) ;

obj.calculation() ;

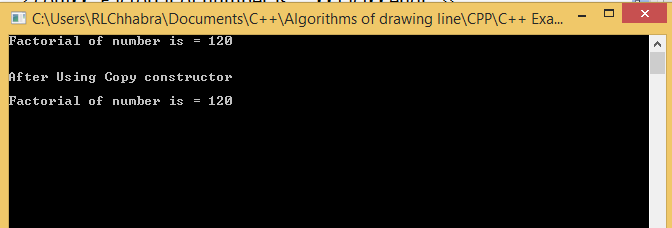
obj.display() ; factorial obj1 = obj ; cout<<”\n\nAfter Using Copy constructor\n”<<endl;

obj1.calculation() ; obj1.display();

getch(); return 0 ; }

**Input Given:** n = 5

**Output:**

****

Experiment 4.b

Date: 19-Sep-2018

**Aim:** To write a c++ program to demonstrate example of friend function using class.

**Algorithm:**

1: Start.

2: Declare a class as Area with variable radius.

3: Create a by default constructor.

4: Create a constructor to initialize radius.

5: Declare a friend function as calculation().

6: Define outside of the class.

7: Definition of main() function.

8: Create an object of class Area.

9: Call calculation using class object.

10: Stop.

**Program:**

#include<conio.h>

#include<iostream>

using namespace std ;

class Area

{ int radius ;

public:

Area() { }

Area(int x )

{ radius = x ; }

friend int calculation(Area);};

int calculation(Area a)

{ return 3.14\*a.radius\*a.radius ; }

int main()

{ Area a(5);

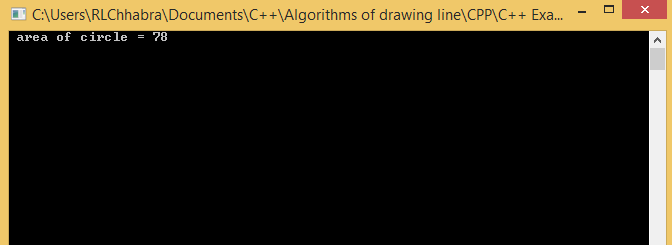
cout<<" area of circle = "<<calculation(a);

getch();

return 0 ; }

**Input Given:** radius = 5

**Output:**

****

Experiment 4.c

Date:19-Sep-2018

**Aim:** To write a cpp program to find square and cubic values of a number using inline function.

**Algorithm:**

1: Start.

2: Define inline functions as Square() and cube().

3: Definition of main() function.

4: Declare a variable n and take value from user.

5: Call function square() and cube() to print result.

6: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

inline int square(int a)

{

return a\*a ;

}

inline int cube(int b)

{

return b\*b\*b ;

}

int main()

{ int n ;

cout<<"\nEnter any number"<<endl;

cin>>n ;

cout<<"\nSquare is = "<<square(n)<<endl;

cout<<"\ncube of number = "<<cube(n)<<endl;

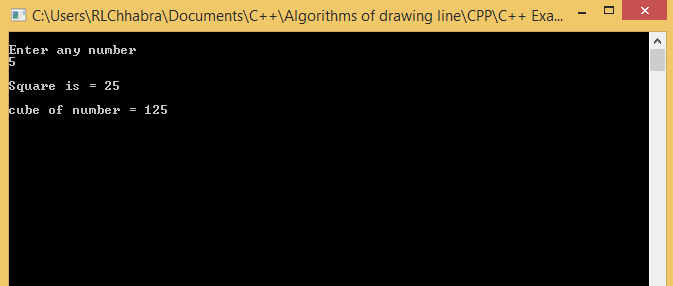
getch();

return 0 ;

}

**Input Given:** n = 5

**Output:**

****

Experiment 4.d

Date: 19-Sep-2018

**Aim:** To write a cPP program to calculate area of circle , rectangle and triangle using function overloading.

**Algorithm:**

1: Start.

2: Declare a class as Area.

3: Define member functions as area() with different type and numbers of arguments.

4: Definition of main().

5: Create a object.

6: Call area() function for circle , rectangle and triangle by using object.

7: Display result.

8: Stop.

**Program:**

#include<iostream>

#include<conio.h>

using namespace std ;

class Area

{ public: float area(int r)

{ return 3.14\*r\*r;}

float area(int x , int y )

{ return x\*y ; }

float area(int a , int b , float x)

{ return a\*b\*x ; } };

int main()

{ Area obj ;

cout<<" \n\narea of circle = "<<obj.area(5)<<endl;

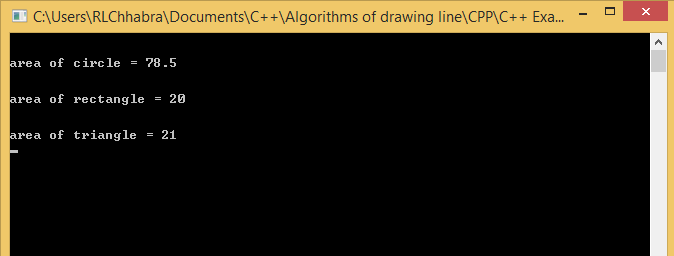
cout<<" \n\narea of rectangle = "<<obj.area(4,5)<<endl;

cout<<" \n\narea of triangle = "<<obj.area(6,7,0.5)<<endl;

getch(); return 0 ; }

**Input Given:** area(5) , area(4,5) , area(6,7,0.5)

**Output:**

****

Experiment 4.e

Date: 19-Sep-2018

**Aim:** To write a c++ program to get detail of students and print using class.

**Algorithm:**

1: Start.

2: Declare a class as student with variable roll , name , year , branch.

3: Define member function as get\_data() to take input from user.

4: Define member function display() to display details of students.

5: Definition of main() function.

6: Declare a variable n and take value from user.

7: Create n objects of class in array.

8: for i <n

obj[i].get\_data()

9: for i<n

obj[i].display()

10: Stop.

**Program:**

#include<iostream>

#include<conio.h>

#include<string>

#include<stdlib.h>

using namespace std ;

class student

{ private: int roll ; string name ; float year ; string branch ;

public: void get\_data()

{ cout<<"Enter name of student"<<endl;

fflush(stdin);

getline(cin,name);

cout<<"Enter name of branch of student"<<endl;

getline(cin,branch);

cout<<"ENter roll number of student"<<endl;

cin>>roll;

cout<<"Enter year of student in which he study"<<endl;

cin>>year ; }

void display()

{ cout<<"name of student is : "<<name<<endl;

cout<<"roll number of student is : "<<roll<<endl;

cout<<"branch is : "<<branch<<endl;

cout<<" year is : "<<year<<endl } };

int main()

{ int n ;

cout<<"Enter Number of studets"<<endl;

cin>>n;

student obj[n];

for(int i =0 ; i<n ; i++)

{ cout<<"Enter details of "<<i+1<<" student"<<endl;

obj[i].get\_data();}

for(int i=0 ; i< n ; i++)

{ cout<<"Details of "<<i+1<<" student\n\n"<<endl;

obj[i].display();}

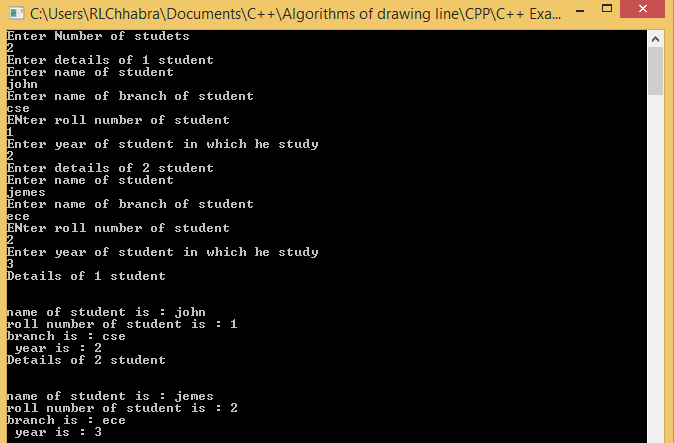
getch();

return 0 ; }

**Input Given:** n = 2 for first student name = “john” , roll = 1 , year = 2 , branch = “cse”

For second student name = “james” , roll = 2 , year = 3 , branch = “ece”

**Output:**

****