1] a] Write a C++ program to create a calculator

→

```
#include <iostream>
using namespace std;
int main() {
  char op;
  int n1, n2;

  cout << "Enter operator:" << endl;
  cin >> op;
  cout << "Enter two operands:";
  cin >> n1 >> n2;
 switch (op) {
  case '+': cout << n1 << "+" << n2 << "=" << n1+n2 << endl;
            break;
  case '-': cout << n1 << "-" << n2 << "=" << n1-n2;
            break;
  case '*': cout << n1 << "*" << n2 << "=" << n1*n2;
            break;
  case '/': cout << n1 << "/" << n2 << "=" << n1/n2;
            break;
  default: cout << "Error! Not a correct operator.";
            break;
  }
  return 0;
}
```

1] b] Write a C++ program to convert seconds into hours, minutes and seconds.

```
#include <iostream>
using namespace std;

void conl(float hrs){
  double min, sec;
  min = hrs * 60;
  sec = hrs * 3600;
  cout<<"There are"<< min <<"minutes in"<< hrs<<"hours";
  cout<<"There are"<< sec <<"seconds in" << hrs<<"hours";
}

int main(){
  float hours;
  cout<<"Enter hours:";
  cin>>hours;

  Conl(hours);

  return 0;
}
```

1. c) Write a C++ program to find the volume of a square, cone and rectangle.

→

```
#include <iostream>
using name space std;

void Sqvol (float side){
    cout<<"The volume of square is "<< side << endl;
  }
void conevol ( float r, float h){
   cout<<"The volume of cone is" << 3.14*r*r*h /3 <<endl;
  }
void recvol ( float l, float w, float h){
 cout <<"The volume of rectangle is" << l*w*h <<endl;
}
int main() {
 float side, r, h1, l, w, h2;
 cout<<"Enter side to find volume of square:" <<endl;
 cin>> side;
 Sqvol(side);
 cout<<"Enter radius and height to find volume of cone:" <<endl;
 cin>> r >> h1;
 conevol (r, h);
 cout<<"Enter lenght, width and height to find volume of
        rectangle:" <<endl;
 cin >> l >> w >> h2;
 recvol (l,w, h2);
 return 0;
}
```

Q.2. a. Write a C++ program to find the greatest of three numbers.

→

```
#include <iostream.h>
using namespace std;

int main() {

    double n1, n2, n3;
    cout << "Enter three numbers: ";
    cin >> n1 >> n2 >> n3;

  if (n1 >= n2 && n1 >= 3) {
    cout << "Largest number: " << n1;
    }
  else if (n2 >= n1 && n2 >= n3) {
      cout << "Largest number: " << n2;
  }
  else {
      cout << "Largest number: " << n3;

  return 0;
}
```

2.b] Write a C++ program to find the sum of even and odd natural numbers.

```
#include <iostream>

int main(){
   int num, max, evSum=0, oddSum=0;

   cout<<"\n Please Enter the Maximum Limit for Even & 
         odd Numbers = ";
   cin >> maximum;   for (num=1; num<=max; num++){
   if (num%2==0){
         evSum = evSum + num;
      }
   else {
      oddsum = oddsum + num;
      }
   }
   cout<<"\n The Sum of All Even Numbers upto"<< max <<
      "=" << evSum;
   cout<<"\n The Sum of All odd Numbers upto"<< max <<
      "=" << oddSum;

      return 0;
}
```

2 c] Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

→

```
#include <iostream>
int main () {
    int num,i , upto;
cout << " Find prime number upto: ";
cin >> upto;

cout <<"All prime numbers upto " << upto << "are:" << endl;

for (num=2; num<= upto; num++) {
    for (i=2; i<= (num/2); i++) {
        if (num % i == 0) {
            i = num;
            break;
        }
}
    if (i != num) {
        cout << num << " ";
    }
}

return 0;
}
```

3] a. Write a program using classes and object student to print name of the student, roll_no. Display the same.

→

```
#include <iostream>
using namespace std;
#define MAX 10;
class student {
private:
    char name[30];
    int rollNo;
    int total;
    float perc;
public:
    void getDetails(void);
    void putDetails(void);
};

void student::getDetails(void){
    cout << "Enter name:";
    cin >> name;
    cout << "Enter roll no.:";
    cin >> rollNo;
    cout << "Enter total marks out of 500:";
    cin >> total;
    //cout <<
    perc = (float) total / 500 * 100;
}
```

```
void student::putDetails(void){
    cout<<"Student details:\n";
    cout<<"Name:"<<name<<",\n Roll number:"<<rollNO
    <<"\n Total:"<<total<<"\n Percentage:"<<per<<endl;
}

int main() {
    student std[MAX];
    int n, loop;

    cout<<"Enter total number of students:";
    cin>>n;

    for(loop=0; loop<n ; loop++){
        cout<<"Enter details of of student "<<loop+1<<endl;
        std[loop].getDetails();
    }
    cout<<endl;

    for(loop=0; loop<n; loop++){
        cout<<"Detail of student"<<(loop+1)<<endl;
        std[loop].putDetails();
    }

    return 0;
}
```

3. d]

```
#include<iostream>
using namespace std;

class a;
class b{
    int num;
  public:
    b(int x) {
        num = x;
    }
  void friend greatest(a a1, b b1);
};};
class a{
    int num;
  public:
   a(int x){
        num = x;
    }
  void greatest(a a1, b b1);
};
```

```
void greatest ( a a1, b b1) {
    if (a1.number > b1.number){
        cout<< a1.num << " is greatest number" <<endl;
    } else if (a1.num < b1.num) {
        cout<< b1.num <<" is greatest number "<<endl;
    } else {
        cout<<" Both numbers are equal";
    }
}

int main(){
    greatest (1, 2);
        cout <<endl;
    return 0;
}
```

3.e]

```
#include <iostream>
using namespace std;

class Point {
    int x,y;
  public:
    Point (int x1, int y1){
        x = x1;
        y = y1;
    }
    Point (const Point & p1){
        x = p1.x;
        y = p1.y;
    }
    int getx() { return x; }
    int gety() { return y; }
};
int main (){
    point p1(10,15);
    point P2 = p1;

    cout << "original number=" << p1.getx() << "original number=" << p1.get
    cout << "copy number=" << p2.getx() << "copy number=" << p2.gety

    return 0;
}
```


4.a]

```
#include<iostream.h>
class Complex{
  int real, imag;
public:
   Complex(int r=0, int i=0){
        real =r;
       imag= i;
    }
  Complex operator+(Complex obj)
  {
    Complex res;
    res.real = real + obj.real;
    res.imag = imag + obj.imag;
    return res;
  }
  void print(){ cout<< real<<"+i"<< imag<<"\n"; }
};          //cout<<imag<<"+i"<<endl;
int main(){

    Complex c1(10,5), c2(2,4);
    Complex c3= c1+c2;
    c3.print();
}
```

4.b]

```
#include<iostream>
#include<stdlib.h>
using namespace std;

class Student{
    string name;
    int age;
  public:
    Student(){
        cout<<"Constructor is called\n";
    }
    Student(string name, int age){
        this->name= name;
        this->age = age;
    }
    void display(){
    cout<<"Name:"<<name<<endl;
    cout<<" Age:"<< age<<endl;
    }
// void *p = ::operator new(size);
 void* operator new(size_t size){
        cout<<" overloading new operator with size:"<<side
 void*p=::operator new(size);
  return p;
}
 void operator delete(void*p){
      cout<<"overloading delete operator"<<endl;
       free(p);
    }
};
```

```
int main() {
    student *p = new student("Pankaj", 18);

    p-> display();
    delete p;
}
```

4.c]

```
#include<iostream>
using namespace std;

class My_Class {
        int num;
    public:
        void set_number(int value){
                num = value;
        }
        void show_number();
};
void My_Class :: show_number(){
        cout<< num <<"\n"; }
int main(){

    My_Class object, *p;

    object.set_number(1);
    // object.set;
    object.show_number();

    p = & object;

    p -> show_number();

    return 0;
}
```

4.e] Single Inheritance

```
#include<iostream>
#include<string>
using namespace std;
class Animal{
    string name = " ";
    public:
      int tail=1;
      int legs = 4;
};
class Dog::public Animal{
   public:
      void voice Action(){
          cout << " Barks!!!";
};  }
 int main(){
     Dog dog;
     cout<<"Dog has"<<dog.legs<<" legs"<<endl;
     cout<<"Dog has"<<dog.tail<<"tail"<<endl;

     cout << "Dog";

     dog.voice Action();

 }
```

4-f] Multiple inheritance

```
#include<iostream>
using namespace std;
class student_marks {
  int rollNo, marks1, marks2;
  public:
    void get() {
        cout << "Enter the Roll No.:";
        cin >> roll no;
        cout<<"Enter the two highest marks:";
        cin >> marks1 >> marks2;
    }
};

class cocurricular_marks {
    int co marks;
      public:
        void getsm() {
            cout<< "Enter the marks for cocurricular
                    Activities:";
            cin >> comarks;
        }
};
class Result : public student_marks, public co-curricular_
marks {
    int total_marks, avg_marks;
```

```
public:
void display(){
    total_marks = (marks1 + marks2 + comarks);
    avg_marks = total_marks / 3;

    cout<<"\n Rollno:" << roNo<<"\n Total marks:"
    << total_marks;
    cout <<"\n Average marks:" << avg_marks;
    }
};

int main(){

    Result res;
    res.get();
    res.getsm();
    res.display();

}
```

4.g] Multi level inheritance

```
#include <iostream>
#include <string>
using namespace std;
class Animal {
    string name = " ";
    public:
    int legs = 4;
    int legs = 1;
};
class Dog : public Animal {
    public:
    void voiceAction(){
        cout << "Barks!!!" << endl;
    }
};
class Puppy : public Dog {
    public:
    void weeping() {
        cout << "Weeps!!";
    }
};
int main(){
    Puppy puppy;
    cout << "Puppy has" << pupy.legs << "legs" << endl;
    cout << "Puppy has" << puppy.tail << "tail" << endl;
    cout << "Puppy"; puppy.voiceAction();
    cout << "Puppy"; puppy.weeping();
}
```

4.b] Hierarchical Inheritance

```
#include<iostream>
using namespace std;

class Shape {
    public:
     int x,y;
    void get_data(int n, int m){
            x=n;
            y=m;
        }
};
class Rectangle : public Shape {
    public:
     int area_rect() {
         int area = x*y;
         return area;
        }
};
class Triangle : public Shape {
    public:
      int triangle_area(){
          float area = 0.5*x*y;
          return area;
        }
};
```

```
class Square : public Shape {
    public:
      int square_area(){
    float area_float = 4 * x;
            return area;
        }
    };
int main(){
 Rectangle r;
 Triangle t;
 Square s;
int length, breadth, base, height, side;
 cout<<"Enter the lenght and breadth of the rectangle:";<<endl;
 cin >>lenght >> breadth;
 r.get_data (lenght, breath);
int reatarea = r.area_rect();
cout<<"Area of the rectangle"<< rect_area <<endl;
cout<<"Enter the base and height of the trinagle:"<< endl;
cin >> base >> height;
t.get_data (base, height);
float tri_area = t.triangle_area();
cout << Enter the lenght of one side of the square:"<<endl;
 cin >> side;
 s.get_data (side
int sq_area = s.square_area();
 cout << "Area of the square =" << sq_area <<endl;
 return 0;
}
```

6.A.]

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
class person
{
char name[20];
float age;
public:
person(char *s, float a)
{
strcpy(name, s);
age = a;
}
person & person greater(person & x)
{
if (x.age >= age) {
return x;}
else {
return *this;
} }
void display(void)
{
cout << "Name:" << name << "\n"
<< "Age:" << age << "\n";
}
};
```

```
void main()
{
clrscr();
person p1 ("Sweta", 25.0), p2 ("Sadaf", 35.0),
        p3 ("Kubra", 32.0);
person p = p1.greater (p3);
cout << "Elder Person is :\n";
p. display ();

p = p1. greater (p2);
cout <<"\nElder Person is :\n";
p. display ();
getch ();
}
```

7. a)

```
# include <iostream.h>
# include <conio.h>
class Base
{
public:
void display()
{
cout<<"\n Display Base";
}
virtual void show ()
{
cout<<"\n Show Base";
}
};
class Derived:public Base
{
public:
void display ()
{
cout<<"\n Display Derived";
}
void show()
{
cout<<"\n Show Derived";
}
};
void main()
{
```

```
Base B;
Derived D;
Base * bptr;
cout << "\n bptr points to Base\n";
bptr = &B;
bptr -> display ();
bptr -> show ();
cout << "\n\n bptr points to derived\n";
bptr = &D;
bptr -> display ();
bptr -> show ();
getch ();
}
```

8.b]

```
#include <iostream>
#include<fstream>
using namespace std;
int main(){
    cout<<"Example for formatted IO"<<endl;

    cout<<"Default:"<<123<<endl; cout<<123<<endl;
    cout<<"width(5):"<<endl;
    cout.width(5);
    cout<<123<<endl;
    cout<<"width(5) and fill('*'):"<<endl;
    cout.width(5);
    cout.fill('*');
    cout<<123<<endl;
    cout.precision(5);
    cout<<"precision(5)-->"<<123.4567890<<endl;
    cout<<"precision(5)-->"<<9.876543210<<endl;
    cout<<"setf(showpos):"<<endl;
    cout.setf(ios::showpos);
    cout<<123<<endl;
    cout<<"unsetf(showpos):"<<endl;
    cout.unsetf(ios::showpos);
    cout<<123<<endl;

    return 0;
}
```

9. a]

```
#include < iostream >
#include <conio.h>

void test(int x){
    try{
      if (x>0){
         throw x;
    } else{
        throw 'x';
     } catch (int x){
cout<<"Catch a integer and the integer is:" << x << endl;
     } catch (char x){
cout<<"Catch a character and that character is:" << x ; }
    }

void main(){
    clrscr();
    cout<<"Testing multiple catches \n:";

    test(10);
    test(0);
    getch();
}
```