coSlips solutions

Slip 1 – A) Write a C++ program to check maximum and minimum of two integer numbers. (Use Inline function and Conditional operator)

Solution:

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


Inline int max(int a, int b) {
  Return ((a > b) ? a : b);
}
Inline int min(int a, int b) {
   Return ((a < b) ? a : b);
}
Int main() {
   Int a, b;
   Cout << “Enter 2 numbers” << endl;
   Cout << “Number 1: “;
   Cin >> a;
   Cout << “Number 2: “;
   Cin >> b;
   Cout << “The maximum number is: “ << max(a, b) << endl;
   Cout << “The minimum number is: “ << min(a, b) << endl;
```

```
  Return 0;
}
```

Slip 1 – B) Create a C++ class MyFile with data membersfile pointer and filename. Write necessary member functions to accept and display File. Overload the following operators: Operator Example Purpose + F3=F1+F2 Concatenate the contents of file F1 and F2 in F3. ! !F3 Changesthe case of alternate characters of file F3.

Solution:

```
#include <stdio.h>
#include <stdlib.h>
#include <istream>
#include <conio.h>
#include <ctype.h>
#include <fstream>
#include <iostream>

Using namespace std;

#include <string.h>
#define MAXSIZE (10)

Class myfile
{
  FILE *fp;
  Char fn[MAXSIZE];
```

```
Public:
   Myfile(const char *fname)
   {
      Strcpy(fn, fname);
   }
   Myfile operator+(myfile);
   Void operator!();
   Void display();
};
Void myfile::display()
{
   Fp = fopen(fn, “r”);
   Char ch;
   While ((ch = fgetc(fp)) != EOF)
   {
      Cout << ch;
   }
   Fclose(fp);
}
Void myfile::operator!()
{
   Myfile f4(“sy.txt”);
   Char ch;
   Fp = fopen(fn, “r”);
   F4.fp = fopen(f4.fn, “w”);
   While ((ch = fgetc(fp)) != EOF)
   {
      If (isupper(ch))
         Fputc(tolower(ch), f4.fp);
```

```
      Else if (islower(ch))
         Fputc(toupper(ch), f4.fp);
      Else
         Fputc(ch, f4.fp);
   }
   Fclose(fp);
   Fclose(f4.fp);
   Remove(“abc.txt”);
   Rename(“sy.txt”, “abc.txt”);
}
Myfile myfile::operator+(myfile f2)
{
   Myfile f3(“abc.txt”);
   Fp = fopen(fn, “r”);
   F2.fp = fopen(f2.fn, “r”);
   F3.fp = fopen(f3.fn, “w”);
   Char ch;
   While ((ch = fgetc(fp)) != EOF)
   {
      Fputc(ch, f3.fp);
   }
   Fclose(fp);
   While ((ch = fgetc(f2.fp)) != EOF)
   {
      Fputc(ch, f3.fp);
   }
   _fcloseall();
   Return f3;
}
```

```
Int main()
{
   Myfile f1(“xyz.txt”);
   Myfile f2(“lmn.txt”);
   Myfile f3(“abc.txt”);

   Cout << “first file \n”;
   F1.display();
   Cout << “\nsecond file \n”;
   F2.display();
   F3 = f1 + f2;
   Cout << “\nAfter concatnation file is “;
   F3.display();
   Cout << “\nAfter changes case \n”;
   !f3;
   F3.display();

   Return 0;
}
```

PHP

Slip 1 – A) Write a PHP script to create a simple calculator that can accept two numbers and perform operations like add, subtract, multiplication. (Use the concept of Class)

Solution:

```
<?php
Class MyCalculator {
```

```
Private $_fval, $_sval;
Public function __construct( $fval, $sval ) {
$this->_fval = $fval;
$this->_sval = $sval;
}
Public function add() {
Return $this->_fval + $this->_sval;
}
Public function subtract() {
Return $this->_fval - $this->_sval;
}
Public function multiply() {
Return $this->_fval * $this->_sval;
}
Public function divide() {
Return $this->_fval / $this->_sval;
}
}
$mycalc = new MyCalculator(12, 6);
```

```
Echo $mycalc-> add().”\n”; // Displays 18

Echo $mycalc-> multiply().”\n”; // Displays 72

Echo $mycalc-> subtract().”\n”; // Displays 6

Echo $mycalc-> divide().”\n”; // Displays 2

?>
```

Slip 1 – B) Write a PHP script to create student.xml file which contains studentroll no, name, address, college and course. Print students detail of specific course in tabular format after accepting course as input.

Solution:

Slip1-p1-q2.html

```
<html>

<script type=”text/javascript”>

Function display()

{

        Name=f1.txt.value;

        Var xmlhttp;

        If(window.XMLHttpRequest)

        {

                Xmlhttp= new XMLHttpRequest();
```

```
        }
        Else
        {
            Xmlhttp=new
ActiveXObject("Microsoft.XMLHTTP");
        }
    Xmlhttp.open("GET", "slip1-p2-q2.php?txt="+
name,false);
    Xmlhttp.send();
    Document.getElementById('result').innerHTML=
            Xmlhttp.responseText;
}
</script>
<body>
        <form name="f1">
            Enter Employee name:
            <input type="text" name="txt"
onKeyUp="display()"/><br>
        </form>
         <div id='result'></div>
```

```
</body>
</html>
```

Slip1-p1-q2.xml

```
<?xml version=”1.0”?>
<Movie_Store>
<Movie>
<Category>Biography</Category>
<MovieName>Dangal</MovieName>
<ReleaseYear>2016</ReleaseYear>
</Movie>
<Movie>
<Category>Action</Category>
<MovieName>Tanhaji</MovieName>
<ReleaseYear>2020</ReleaseYear>
</Movie>
<Movie>
<Category>Family</Category>
<MovieName>The Sky is Pink</MovieName>
```

```
<ReleaseYear>2019</ReleaseYear>
</Movie>
</Movie_Store>
```

Slip1-p2-q2.php

```
<?php
$msearch=$_GET['txt'];
$xml=simplexml_load_file("slip1-p2-q2.xml");
Echo $xml->getName()."<br>";
$hint="";
$len=strlen($msearch!=="");
If($msearch !=="")
{
	Foreach($xml->children() as $student)
	{
		//if(strstr($movie-
>MovieName,substr($msearch,0,$len)))
		If(strcmp($msearch,$student->sname)==0)
		{
```

```
                $hint=" Student Name: $student->sname <br>
                    Cname: $student->cname<br>
                    Percentage:$student->percentage<br>";
            }
        }

}
 Echo $hint===""? "no suggestion":$hint;
 ?>
```

Slip1-p2-q2.xml

```
<Course>
<SYBBACA>
<sname>Harsh</sname >
<cname>SYBBACA</cname >
<percentage>82</percentage>
<sname>Yash</sname >
<cname>SYBBA</cname >
<percentage>90</percentage>
```

<sname>Aman</sname >

<cname>SYBBAIB</cname >

<percentage>67</percentage>

<sname>Ajay</sname >

<cname>FYBBAIB</cname >

<percentage>77</percentage>

<sname>Raj</sname >

<cname>FYBBA</cname >

<percentage>87</percentage>

</SYBBACA>

</Course>

SLIP 2

Slip 2 – A) Write a C++ program to find volume of cylinder, cone and sphere. (Use function overloading).

Solution:

#include<iostream>

Using namespace std;

Float volume(float r, float h) {

```
    Return (3.14*r*2*h);
}

Float coneVol(float r, float h) {
    Return (3.14*r*2*h/3);
}

Float volume(float r) {
    Return (4/3*3.14*r*r*r);
}


Int main() {
    Float cy_h, cy_r, co_h, co_r, sp_r;

    Cout << “Enter dimensions” << endl;

    Cout << “1. Cylinder” << endl;
    Cout << “Height: “;
```

```
Cin >> cy_h;
Cout << “Radius: “;
Cin >> cy_r;

Cout << endl;

Cout << “2. Cone” << endl;
Cout << “Height: “;
Cin >> co_h;
Cout << “Radius: “;
Cin >> co_r;

Cout << endl;

Cout << “3. Sphere” << endl;
Cout << “Radius: “;
Cin >> sp_r;

Cout << endl;
```

```
   Cout << “The volume of Cylinder is: “ << volume(cy_h, cy_r) << endl;
   Cout << “The volume of Cone is: “ << coneVol(co_h, co_r) << endl;

   Cout << “The volume of Sphere is: “ << volume(sp_r) << endl;

   Return 0;
}
```

PHP.....

Slip 2 – A) Write a PHP script to demonstrate the introspection for examining classes and objects. (use function get_declared_classes() ,get_class_methods() and get_class_vars()).

Solution:

```
<?php
Class Myclass
{
        Public $a;
```

```
        Public $b=305;
        Public $c='Akshay';
        Function Myclass()
    {
            //Myclass function
        }
        Function myfun1()
    {
            //functin
    }
        Function myfun2()
    {
            //functin
    }
}
        $class=get_declared_classes();
        Foreach($class as $cname)
        {
            Echo"$cname<br>";
```

```
}
Echo”<br>Class Methods are : <br>”;
$m=get_class_methods(‘Myclass’);
Foreach($m as $mname)
{
        Echo”$mname<br>”;
}
$cp=get_class_vars(‘Myclass’);
Echo”class variables are :<br>”;
Foreach($cp as $cpname => $v)
{
        Echo”$cpname : $v <br>”;
}

?>
```

Slip 3 – A) Write a C++ program to interchange values of two integer numbers. (Usecall by reference).

Solution;

```
#include<iostream>
```

```
Using namespace std;

Void swap(int &a, int&b) {
   Int temp;

   Temp = a;
   A = b;
   B = temp;
}

Int main() {
   Int a,b;

   Cout << “Enter two numbers” << endl;

   Cout << “Enter first number: “;
   Cin >> a;
   Cout << “Enter second number: “;
```

```
   Cin >> b;

   Cout << endl << “Original Numbers” << endl << “a = “ << a << “ & b = “ << b << endl;

   Swap(a, b);

   Cout << endl << “After swap Numbers” << endl << “a = “ << a << “ & b = “ << b << endl;

   Return 0;
}
```

Slip 3 – B) Write a C++ program to accept ‘n’ numbers from user through Command Line Argument. Store all Even and Odd numbers in file “Even.txt” and “Odd.txt” respectively.

Solution:

```
#include <stdio.h>
#include <stdlib.h>
#include <istream>
#include <conio.h>
```

```
#include <ctype.h>
#include <fstream>
#include <iostream>

Using namespace std;

Int main(int argc, char *argv[])
{

   Int num;
   Ofstream f1(“even.txt”),f2(“odd.txt”);



   If (argc == 1)
      Printf(“No command line arguments found.\n”);

   Else
   {
      For(int I = 1; I < argc; i++)
```

```
    {
      Num = atoi(argv[i]);
      If(num%2==0)
      {
        F1<<num;
      }
      Else
      {
        F2<<num;
      }
    }
  }
  Return 0;
}
```

PHP

Slip 3 – A) Write a Calculator class that can accept two values, then add, subtract, multiply them or divide them on request. For example: $calc = new Calculator(3, 4); echo $cale-

>add(); // Displays “7” echo $calce- >multiply(); // Displays “12”

Solution:

```
<?php
Class Calculate
    {public $a;
Public $b;
Function __construct($a,$b){
$this->a=$a;
$this->b=$b;
}
Public function add()
      {
       $c=$this->a+$this->b;
Echo”Addition = $c<br>”;
      }
Public function subtract()
      {
       $c=$this->a-$this->b;
```

```
Echo”Subtract = $c<br>”;
        }
Public function multiply()
        {
         $c=$this->a*$this->b;
Echo”Multiplication = $c<br>”;
        }
Public function div()
        {
         $c=$this->a/$this->b;
Echo”Division = $c”;
        }}
//$x=$_GET[‘a’];
//$y=$_GET[‘b’];
$calc=new Calculate(4,3);
$calc->add();
$calc->subtract();
$calc->multiply();
$calc->div();?>
```

Slip 3 – B) Write a script to create “cricket.xm1” file with multiple elements as shown below:

Solution:

<CricketTeam>

<Team country=”India”>

<player>ROHIT SHARMA</player>

<runs>2850</runs>

<wicket>5</wicket>

</Team>

<Team country=”England”>

<player>Ben Stokes</player>

<runs>2650</runs>

<wicket>25</wicket>

</Team>

<Team country=”Australia”>

<player>Steve Smith</player>

<runs>7000</runs>

<wicket>20</wicket>

</Team>

<Team country=”West Indies”>

<player>DJ BRAVO</player>

<runs>9000</runs>

<wicket>225</wicket>

</Team>

<Team country=”Bangladesh”>

<player>Shakib-al-Hasan</player>

<runs>4190</runs>

<wicket>135</wicket>

</Team>

</CricketTeam>

Slip 4 – A) Write a C++ program to accept Worker information Worker_Name, No_of_Hours_worked, Pay_Rate and Salary. Write necessary functions to calculate and display the salary of Worker. (Use default value for Pay_Rate)

Solution:

#include<iostream>

Using namespace std;

```
Class WorkerInformation {
   Char Worker_Name[50];
   Int No_Of_Hours_Worked, Salary;

   Public:
      Void acccept() {
         Cout << “Enter name of the worker: “;
         Cin >> Worker_Name;

         Cout << “Enter number of hours worked: “;
         Cin >> No_Of_Hours_Worked;


      }

      Void display() {
         Cout << endl << “Worker details” << endl;
         Cout << “Name: “ << Worker_Name << endl;
```

```
        Cout << “Salary: “ << calSal(No_Of_Hours_Worked) << endl;
    }

    Int calSal(int work_hrs, int pay_rate=100) {
        Return (work_hrs*pay_rate);
    }
};


Int main() {
  WorkerInformation w;

  w.acccept();
  w.display();
  return 0;
}
```

Slip 4 – B) Write a C++ program to create a base class Employee (Emp-code, name, salary). Derive two classes as Fulltime (daily_wages, number_of_days) and Parttime

(number_of_workinghours, hourly_wages). Write a menu driven program to perform following functions: 1. Accept the details of ‘n’? employees and calculate the salary. 2. Display the details of ‘n’ employees. 3. Display the details of employee having maximum salary for both types of employees.

Solution:

```
#include<iostream>
Using namespace std;
Const int m=50;

Class emp
{
    Public:
        Int empcode;
        Char empname[30];
        Int salary;
    Public:
        Void get()
        {
```

```
            Cout<<"\n Enter Employee No.  : ";
            Cin>>empcode;
            Cout<<"\n Enter Employee Name  : ";
            Cin>>empname;

        }
};
Class fulltime:public emp
{
    Public:
        Float daily_wages;
        Int no_of_days;

    Public:
        Void getdata()
        {
            Cout<<"\n Enter Daily Rate    : ";
            Cin>>daily_wages;
            Cout<<"\n Enter No. of Days   : ";
```

```
        Cin>>no_of_days;
    }
    Void cal()
    {
        Salary=daily_wages*no_of_days;
        Cout<<"\n Salary          : "<<salary;
    }
    Void show()
    {
        Cout<<"\n --------------------------------\n";
        Cout<<"\n Employee Number   : "<<empcode;
        Cout<<"\n Employee Name     : "<<empname;
        Cout<<"\n Salary          : "<<salary;
        Cout<<"\n Status          : Fulltime";
        Cout<<"\n --------------------------------\n";
    }
};
Class parttime:public emp
{
```

```
    Public:
        Int hourly_wages;
        Int working_hours;

    Public:
        Void get1()
        {
            Cout<<"\n Enter Hourly Rate   : ";
            Cin>>hourly_wages;
            Cout<<"\n Enter Working Hours  : ";
            Cin>>working_hours;
}
Void cal1()
{
            Salary=hourly_wages*working_hours;
            Cout<<"\n Salary          : "<<salary<<endl;
        }
        Void show1()
        {
```

```
            Cout<<"\n ---------------------------------\n";
            Cout<<"\n Employee No    : "<<empcode;
            Cout<<"\n Employee Name  : "<<empname;
            Cout<<"\n Salary        : "<<salary;
            Cout<<"\n Status        : Part time";
            Cout<<"\n ---------------------------------\n";
        }
};
Int main()
{

    Int const cnt=5;
    Int var=0;
    Int var1=0;
    Fulltime f1[cnt];
    Parttime p1[cnt];

    Int x,I;
    Do
```

```
{
    Cout<<"\n";
    Cout<<"\n 1.Enter Record";
    Cout<<"\n 2.Display Record";
    Cout<<"\n 3.Search Record";
    Cout<<"\n 4.Quit";
    Cout<<"\n\n Enter Your Choice : ";
    Cin>>x;
    Switch(x)
    {
        Case 1:
            Int y;
            Cout<<"\n 1. Fulltime Employee";
            Cout<<"\n 2. Parttime Employee \n";
            Cout<<"\n Enter : ";
            Cin>>y;
            Switch(y)
            {
                Case 1:
```

```
                F1[var].get();
                F1[var].getdata();
                F1[var].cal();
                Var++;
                Break;
            Case 2:
                P1[var1].get();
                P1[var1].get1();
                P1[var1].cal1();
                Var1++;
                Break;
        }
        Break;
Case 2:
        For(i=0; i<var; i++)
        {
            F1[i].show();
        }
        For(i=0; i<var1; i++)
```

```
        {
            P1[i].show1();
        }
Break;
Case 3:
        Int a;
        Cout<<"\n Enter Employee No. : ";
        Cin>>a;
        For (int i=0; i<var; i++)
        {
            If (f1[i].empcode==a)
            {
                F1[i].show();
            }
            If(p1[i].empcode==a)
            {
                P1[i].show1();
            }
        }
```

```
        }

    } while(x!=4);
Return 0;
}
```

PHP

Slip 4 – A) Define a class Employee having private members — id, name, department, salary. Define parameterized constructors. Create a subclass called “Manager” with private member bonus. Create 3 objects of the Manager class and display the details of the manager having the maximum total salary (salary + bonus).

Solution:

```
<?php
Class Employee
{
Private $eid,$ename,$edept,$sal;
Function __construct($a,$b,$c,$d)
{
```

```
$this->eid=$a;
$this->ename=$b;
$this->edept=$c;
$this->sal=$d;
}
Public function getdata()
{
Return $this->sal;
}
Public function display()
{
Echo $this->eid.”
“;
Echo $this->ename.”
“;
Echo $this->edept.”
“;
}
}
```

```
Class Manager extends Employee
{
Private $bonus;
Public static $total1=0;
Function __construct($a,$b,$c,$d,$e)
{
Parent::__construct($a,$b,$c,$d);
$this->bonus=$e;
}
Public function max($ob)
{
$sal=$this->getdata();
$total=$sal+$this->bonus;
If($total>self::$total1)
{
Self::$total1=$total;
Return $this;
}
Else
```

```
{
Return $ob;
}
}
Public function display()
{
Parent::display();
Echo self::$total1;
}

}
$ob=new Manager(0,”ABC”,””,0,0);
$ob1=new Manager(1,”ramdas”,”computer”,28000,2000);
$ob=$ob1->max($ob);
$ob2=new Manager(2,”ramdas1”,”computer”,30000,2500);
$ob=$ob2->max($ob);
$ob3=new Manager(3,”ramdas2”,”computer”,32000,3000);
$ob=$ob3->max($ob);
$ob4=new Manager(4,”ramdas”,”computer”,28000,4000);
```

```
$ob=$ob4->max($ob);

$ob5=new Manager(5,”ramdas”,”computer”,28000,5000);

$ob=$ob5->max($ob);

$ob->display();

?>
```

Slip 4 – B) Create an xml file which should comprise the following: Sachin Tendulkar 2000 100 20 Forat least 5 players. Write a PHPscript to display the details of players who have scored more than 1200 runs and atleast 50 wickets.

Solution:

Slip4-p2-q2.php

```
<?php

$d=new DOMDocument();

$d->load(“slip4-p1-q2.xml”);


$run=$d->getElementsByTagName(‘runs’);

$wic=$d->getElementsByTagName(‘wickets’);

$name=$d->getElementsByTagName(‘player’);


Foreach($name as $n)
```

```
{
        If($run>=’12*00’ && $wic>=’50’)
        Echo “<br>”.$n->textContent;
        Else “not”;
}
?>
```

Slip4-p1-q2.xml

```
<?xml version=’1.0’ encoding=’UTF-8’?>
<cricketinfo>
        <cricket>
                <player>abc</player>
                <runs>1000</runs>
                <wickets>50</wickets>
                <noofnotout>10</noofnotout>
        </cricket>

        <cricket>
                <player>def</player>
```

```
        <runs>100</runs>
        <wickets>40</wickets>
        <noofnotout>10</noofnotout>
</cricket>

<cricket>
        <player>pqr</player>
        <runs>1020</runs>
        <wickets>60</wickets>
        <noofnotout>10</noofnotout>
</cricket>

<cricket>
        <player>xyz</player>
        <runs>9000</runs>
        <wickets>90</wickets>
        <noofnotout>40</noofnotout>
</cricket>
```

```
        <cricket>
                <player>lmn</player>
                <runs>170</runs>
                <wickets>80</wickets>
                <noofnotout>8</noofnotout>
        </cricket>
</cricketinfo>
```

Slip 5 – A) Considerthe following C++ class class Point { int x.y; public: void setpoint(int, int); // To set the values of x and y co-ordinate. Void showpoint(); /| To display co-ordinate of a point P in format(x, y). }

Solution:

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

Class Point {
   Int x, y;

   Public:
```

```
    Void setpoint(int a, int b) {
        X = a;
        Y = b;
    }

    Void showpoint() {
        Cout << “(“ << x << “, “ << y << “)”;
    }
};


Int main() {
    Int a, b;
    Point p;

    Cout << “Enter coordinates” << endl;
    Cout << “Enter x: “;
    Cin >> a;
    Cout << “Enter y: “;
```

```
    Cin >> b;

    p.setpoint(a, b);
    p.showpoint();

    return 0;
}
```

Slip 5 – B) Create a C++ base class Shape. Derive three different classes Circle, Sphere and Cylinder from shape class. Write a C++ program to calculate area of Circle, Sphere and Cylinder. (Use pure virtual function).

Solution:

```
#include
Using namespace std;

Class Shape{
Public:
Virtual void area() = 0;
```

```
};

Class Circale:public Shape{
Public:
Void area(){
Float r;
Cout << “Enter radious: “;
Cin >> r;
Cout << “The area of circale is: “ << 3.14 * r * r << endl;
}
};

Class Sphear:public Shape{
Public:
Void area(){
Float r;
Cout << “\nEnter radious: “;
Cin >> r;
Cout << “\nThe Area of Sphear is: “ << (4.0*3.14*r*r);
```

```
}
};

Class Cylender:public Shape{
Public:
Void area(){
Float r,h;
Cout << “Enter the radious: “;
Cin >> r;
Cout << “Enter the Height: “;
Cin >> h;

Cout << “The area of Sphear is: “ <<2.0*3.14*r*(r+h);
}
};

Int main(){
Shape *ptr;
```

```
Circale obj;
Ptr = &obj;
Ptr->area();

Sphear obj2;
Ptr = &obj2;
Ptr->area();

Cylender obj3;
Ptr = &obj3;
Ptr->area();

Return 0;
}
```

PHP

Slip 5 – A) Create an abstract class Shape with methods area( ) and volume( ). Derive three classes rectangle (length, breath), Circle(radius) and Cylinder(radius, height), Calculate area and volume of all. (Use Method overriding).

Solution:

Slip5-p2-q1.php

```
<?php
Define('pi',3.14);
Abstract class shape
{
 Abstract function calc_area($r,$h);
 Abstract function calc_vol($r,$h);
}
Class circle extends shape
{
  Function calc_area($r,$a)
  {
          Return pi*$r*$a;
  }

}

Class cylinder extends shape
{
```

```
 Function calc_area($r,$h)
 {
        Return 2*pi*$r*($r+$h);
 }

 Function calc_vol($r,$h)
 {
        Return pi*$r*$r*$h;
 }
}

Class rectangle extends shape
{
 Function calc_area($r,$h)
 {
        Return $r*$h;
 }

}
```

```
$r=$_GET[‘r’];
$h=$_GET[‘h’];
$a=$r;
$ob=new rectangle();
Echo “Area of rectangle “.$ob->calc_area($r,$h);
Echo “</br>”;
$ob=new cylinder();
Echo “Area of cylinder “.$ob->calc_area($r,$h);
Echo “</br>”;
Echo “Volume of cylinder”.$ob->calc_vol($r,$h);
Echo “</br>”;
$ob=new circle();
Echo “Area of circle “.$ob->calc_area($r,$a);
Echo “</br>”;
?>
```

Slip5-p1-q1.html

```
<html>
<body>
```

<form action=”slip5-p2-q1.php” method=get>

<h2>Enter values for Cone & Cylinder</h2>

<p>Enter Radius </td><td><input type=”text” name=”r”>

<p>Enter Height</td><td> <input type=”text” name=”h”>

<p><input type=”submit” value=”calculate”>

</form>

</body>

</html>

Slip 5 – B) Create student table as follows: Student(sno, sname, standard, Marks, per). Write AJAX script to select the student name and print the student’s details of particular standard.

Solution:

Html file :

<html>

<body>

<form action=”slip_12.php” method=”get”>

Enter student name Name:<input type=”text” name=sname>

<input type=”submit” value=submit>

</form>

</body>

</html>

Php file :

<?php

$sname=$_GET[‘sname’];

$con=mysql_connect(“localhost”,”root”,””);

$d=mysql_select_db(“bca_programs”,$con);

$result=mysql_query(“select *from student where sname=’$sname’”);

While($row=mysql_fetch_array($result))

{

Echo $row[‘sno’].”—“.$row[‘sname’].”—“.$row[‘per’].”
”;

}

?>

Slip 6 – A) Write a C++ program to create two Classes Square and Rectangle. Compare area of both the shapes using friend

function. Accept appropriate data members for both the classes.

Solution:

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

Class Square {
  Public:
    Int s;

    Void getdata() {
      Cout << “Enter the side of the square: “;
      Cin >> s;
    }

    Int calArea() {
      Return (s*s);
    }
```

```
    Friend void compare(int s, int r);
};

Class Rectangle {
  Public:
    Int l, w;

    Void getdata() {
      Cout << “Enter the length of the rectangle: “;
      Cin >> l;
      Cout << “Enter the width of the rectangle: “;
      Cin >> w;
    }

    Int calArea() {
      Return (l*w);
    }

    Friend void compare(int s, int r);
```

```
};

Void compare(int s, int r) {
   If(s > r) {
      Cout << “The area of square is bigger than area of rectangle.” << endl;
   } else {
      Cout << “The area of rectangle is bigger than area of square.” << endl;
   }
}

Int main() {
   Int s_area, r_area;
   Square s1;
   Rectangle r1;

   S1.getdata();
   S_area = s1.calArea();
```

```
    R1.getdata();
    R_area = r1.calArea();

    Cout << “Square: “ << s_area << endl;
    Cout << “Rectangle: “ << r_area << endl;

    Compare(s_area, r_area);

    Return 0;
}
```

Slip 6 – B) Create a C++ class class Matrix { int **p; intr, ¢; public: // member functions }; Write necessary member functions to: i. Accept Matrix elements ii. Display Matrix elements. Iii, Calculate transpose of a Matrix. (Use constructor and destructor).

Solution:

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

```
Using namespace std;


Class matrix
{
        Int **a,I,j,r,c;
Public:
        Matrix()        // Dynamic Constructor
        {
                Cout<<"Enter Row and Coloum of Matrix :\n";
                Cin>>r>>c;
                A=new int*[r];
                For(i=0; i<r; i++)
                  {
                        A[i]=new int[c];
                  }
        }
        Void accept()
        {
                Cout<<"Enter elements for matrix:\n";
```

```
        For(i=0; i<r; i++)
        {
            For(j=0; j<c; j++)
              {
                  Cin>>a[i][j];
              }
        }
}
Void display()
{
        Cout<<"Elements of matrix are:\n";
        For(i=0; i<r; i++)
        {
            For(j=0; j<c; j++)
              {
                  Cout<<a[i][j]<<"\t";
              }
              Cout<<endl;
        }
```

```
        }
        Void transpose()
        {
                Cout<<"Transpose of Matrix is :\n";
                For(i=0;i<c;i++)
                {
                        For(j=0;j<r;j++)
                        {
                                Cout<<a[j][i]<<"\t";
                        }
                        Cout<<endl;
                }
        }
};

Int main()
{

        Matrix obj1;
```

```
        Obj1.accept();
        Obj1.display();
        Obj1.transpose();

}
```

PHP

Slip 6 – A) Write a PHP script, which will return the following component of the URL (Using response header) http://www.college.com/Science/CS.php List of Components: scheme,host, path Expected output: Scheme: http Host: www.college.com Path: /Science/CS.php

Solution:

```
<?php
$url = 'http://www.college.com/Science/Cs.php';
$url=parse_url($url);
Echo 'Scheme : '.$url['scheme']."<br>";
Echo 'Host : '.$url['host']."<br>";
Echo 'Path : '.$url['path']."<br>";
```

?>

Slip 6 – B) Create employee table as follows EMP (eno, ename,designation, salary). Write Ajax program to select the employees name and print the selected employee's details.

Solution:

Slip6-p1-q2.html

```
<html>
<head>
<script>
Function showUser(str) {
 If (str == “”) {
  Document.getElementById(“txtHint”).innerHTML = “”;
  Return;
 } else {
  Var xmlhttp = new XMLHttpRequest();
  Xmlhttp.onreadystatechange = function() {
   If (this.readyState == 4 && this.status == 200) {
    Document.getElementById(“txtHint”).innerHTML = this.responseText;
   }
```

```
    };
    Xmlhttp.open("GET","slip6-p2-q2.php?q="+str,true);
    Xmlhttp.send();
  }
}
</script>
</head>
<body>

<form>
<select name="users" onchange="showUser(this.value)">
  <option value="">Select a person:</option>
  <option value="harsh">harsh</option>
  <option value="lokya">lokya</option>
  <option value="Aryan">Aryan</option>
  <option value="Yash">Yash</option>
  </select>
</form>
<br>
```

```
<div id=”txtHint”><b>Person info will be listed here...</b></div>

</body>
</html>
```

Slip6-p2-q2.php

```
<?php
$host = “localhost”;
$user = “root”;
$password = “”;
$dbname = “q3”;

$con = mysqli_connect($host, $user, $password,$dbname);


If (!$con) {
  Die(“Connection failed: “ . mysqli_connect_error());
```

```
}
?>
<html>
<head>
<style>
Table {
  Width: 100%;
  Border-collapse: collapse;
}

Table, td, th {
  Border: 1px solid black;
  Padding: 5px;
}

Th {text-align: left;}
</style>
</head>
<body>
```

```
<?php
$q = $_GET[‘q’];
Mysqli_select_db($con,”ajax”);
$sql=”SELECT * FROM emp WHERE name = ‘”.$q.”’”;
$result = mysqli_query($con,$sql);

Echo “<table>
<tr>
<th>Eid</th>
<th>Employeename</th>
<th>Designation</th>
<th>Contact Number</th>
</tr>”;
While($row = mysqli_fetch_array($result)) {
  Echo “<tr>”;
  Echo “<td>” . $row[‘id’] . “</td>”;
  Echo “<td>” . $row[‘name’] . “</td>”;
  Echo “<td>” . $row[‘designation’] . “</td>”;
```

```
  Echo “<td>” . $row[‘number’] . “</td>”;
  //echo “<td>” . $row[‘Job’] . “</td>”;
  Echo “</tr>”;
}
Echo “</table>”;
Mysqli_close($con);
?>
</body>
</html>
```

Slip 7 – A) Write a C++ program using class with data member char str[50] and function replace (char ch 1, char ch2) every occurrence of chl in str should be replaced with ch2 and return number of replacement it makes use default value for char ch2. (Use ch2 as Default Argument)

Solution:

```
#include<conio.h>

#include<iostream.h>
```

```
#include<string.h>

Class mystr

{

Public:int replace(char *str,char,char);

};

Int mystr::replace(char *str,char(),char c2='r')

{

While(str!='\0')

{
```

```
If(*str==c1)

{

*str==c2;

N++;

}

Str++;

I++;

}

Str=str-I;

Return n;
```

```
}

Int main()

{

Mystr m;

Char *str,c1,c2;

Clrscr();

Cout<<”Enter string”;

Cin>>str;

Cout<<”Enter character which is to replace”;
```

```
Cin>>a;

Cout<<"Number of replacement="<<m.replace(str,c1,c2);

Cout<<"After replacement string is ="<<str;

Getch();

Return 0;
```

Slip 7 – B) Create a C++ class Vector with data members size & pointer to integer. The size of the vector varies so the memory should be allocated dynamically. Perform following operations: i. Accept a vector ii. Display a vector in the format (10, 20, 30,....) iii. Calculate union of two vectors. (use parameterized constructor & copy constructor)

Solution:

```
#include<conio.h>

#include<iostream.h>
```

```
Class vector

{

Int *a,*b;

Int n,   n1;

Public:

Void create()

{

Int I,j;

Cout<<”\nEnter the dimensions of the vector space: “;
```

```
Cin>>n;

A=new int[n];

Cout<<"\nEnter First the vector: ";

For(i=0;i<n;i++)

{

Cin>>a[i];

}

Cout<<"\nEnter the dimensions of the vector space: ";

Cin>>n1;

B=new int[n1];
```

```
Cout<<”\nEnter Second the vector: “;

For(j=0;j<n1;j++)

{

Cin>>b[j];

}

}

Void display()

{

Int I,j;
```

```
Cout<<"\n The First vector is: (";

For(i=0;i<n-1;i++)

{

Cout<<a[i]<<",";

}

Cout<<a[n-1]<<")";

Cout<<"\n The Second vector is: (";

For(j=0;j<n1-1;j++)

{

Cout<<b[j]<<",";
```

```
}

Cout<<b[n1-1]<<”)”;

}

 };

 Void main()

{

Vector v;

Clrscr();

Int ch;
```

```
Do

{

Cout<<”\n 1.Accpet vector \n 2.Display Vetor \n Union”;

Cout<<”\n Enter choice : “;

Cin>>ch;

Switch(ch)

{

Case 1 : v.create();

        Break;

Case 2 : v.display();
```

```
        Break;

Case 3:

Vector<int> v(10);

Vector<int>::iterator st;

Sort (a[i], a[i] + 3);//3 element

Sort (b[j], b[j] + 3);

St = set_union(a[i], a[i] + 3, b[j], b[j] + 3, v.begin());

v.resize(st – v.begin());

cout<<"The union between the sets has "<< (v.size())<< " elements: "<<endl;
```

```
for (st = v.begin(); st != v.end(); ++st)

cout<< *st<<” “;

  break;

}

}while(ch!=3);

Getch();

 }
```

PHP

Slip 7 - A) Define an Interface which has method gmtokg() &kgtogm(). Create Class Convert which implements this interface & convert the value kg to gm and gm to kg.

Solution:

slip7-p1-q1.html

```
<html>
<body>
<form action="slip7-p2-q1.php" method="get">

   <h1>Enter weight  in gram</h1>
       <input type="text" name="cel">
       <h1>Enter weight in kg</h1>
       <input type="text" name="far">
       <br><h1>SELECT YOUR CHOICE:<br>
            <input type="radio" name=r value="1">1.Convert to kg<br>
            <input type="radio" name=r  value="2">2.Convert to gram<br></h1>
       <input type="submit" value="calculate">
</form>
</body>
</html>
```

slip7-p2-q1.php

```
<?php
```

```
//$r=$_POST["r"];
//$h=$_POST["h"];
$cs=$_GET["cel"];
$f=$_GET["far"];
$ch=$_GET["r"];
                interface kg
    {
                function convert_to_kg();
                function convert_to_gram();
    }

class Convert implements Kg
{
      public $c,$f;
      function __construct($cs,$f)
      {
        $this->c=$cs;
        $this->f=$f;
```

```
    }

    function convert_to_kg()
    {
        $ans=($this->c/1000);
        echo"Gram into Kilogram conversion:".$ans;
    }
    function convert_to_gram()
    {
        $ans=($this->f*1000);
        echo"Kilogram into gram conversion:".$ans;
    }

}

$cs=$_GET["cel"];
$f=$_GET["far"];
$ch=$_GET["r"];
$obj= new Convert($cs,$f);
```

```
if($ch==1)
{
	$obj->convert_to_kg();
}
if($ch==2)
{
	$obj->convert_to_gram();
}
?>
```

Slip 7 - B) Consider the following relational database: Project (P_Group No, Project_Tiltle) Student (Seat_no, Name, Class, P_Group_No) Write an AJAX script to accept projecttitle and display list of students those who are working in a particular project.

Solution:

slip7-p1-q2.html

```
<html>
<script type="text/javascript">
function display()
{
```

```
        name=f1.txt.value;
        var xmlhttp;
        if(window.XMLHttpRequest)
        {
                xmlhttp= new XMLHttpRequest();
        }
        else
        {
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
                        xmlhttp.open("GET",     "slip7-p2-q2.php?txt="+
name,false);
      xmlhttp.send();
      document.getElementById('result').innerHTML=
      xmlhttp.responseText;
}
</script>
<body>
        <form name="f1">
```

```
        Enter Project_title:
        <input type="text" name="txt" onKeyUp="display()"/><br>
    </form>
     <div id='result'></div>
    </body>
    </html>
```

slip7-p2-q2.php

```
<?php
$host = "localhost";
$user = "root";
$password = "";
$dbname = "q3";

$con = mysqli_connect($host, $user, $password,$dbname);


if (!$con) {
```

```
  die("Connection failed: ".mysqli_connect_error());
}
?>
<html>
<head>
<style>
table
{
  width: 100%;
  border-collapse: collapse;
}

table, td, th {
  border: 1px solid black;
  padding: 5px;
}

th {text-align: left;}
</style>
```

```
</head>
<body>
<?php
$q = $_GET['txt'];
mysqli_select_db($con,"q3");
$sql=("SELECT * FROM project7,student7 WHERE project7.p_group_no=student7.p_group_no and Project_title = '".$q."'");
$result = mysqli_query($con,$sql);

echo "<table>
<tr>
<th>Eid</th>
<th>Employeename</th>
<th>Designation</th>
<th>Contact Number</th>
</tr>";
if(mysqli_num_rows($result)>0)
{
        while($row = mysqli_fetch_array($result))
```

```
{
  echo "<tr>";
  echo "<td>" . $row['Seat_no'] . "</td>";
  echo "<td>" . $row['Name'] . "</td>";
  echo "<td>" . $row['Class'] . "</td>";
  echo "<td>" . $row['Project_title'] . "</td>";
  echo "</tr>";
}
}
echo "</table>";
mysqli_close($con);
?>
</body>
</html>
```

Slip 8 - A) Write a C++ program to create a class Number, which contain static data member 'cnt' and member function 'Display()'. Display() should print number of times display operation is performed irrespective of the object responsible for calling Display() .

Solution:

```
#include <iostream>


using namespace std;
class Number{
   public:
      void display(){
         static int cnt=1;
                  cout<<"\nDisplay function is called "<<cnt<<" times"<<endl;
         cnt++;
      }
};
int main()
{
   Number n1,n2;
   n1.display();
   n1.display();
   n2.display();
   n2.display();
```

```
   return 0;
}
```

Slip 8 - B) Create a C++ class Person with data members Person_name, Mobile_number, Age, City. Write necessary member functions for the following: i. Search the mobile number of given person. ii. Search the person name of given mobile number. iii. Search all person details of given city. (Use function overloading)

Solution:

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

using namespace std;
class person
{
        public:
  int no,mob;
  char name[10],city[10];
  void acc() // function overloading
```

```
{
     cout<<"\nEnter person no : ";
  cin>>no;
  cout<<"\nEnter person name : ";
  cin>>name;
  cout<<"\nEnter person city : ";
  cin>>city;
  cout<<"\nEnter person mob no : ";
  cin>>mob;

}
void acc(char nme[]) // function overloading
{
     if(strcmp(nme,name)==0)
  {
  cout<<"\nPerson name   : "<<name;
  cout<<"\nPerson mob no : "<<mob<<endl;
  }
}
```

```
  void acc(int mno)  // function overloading
  {
       if(mno==mob)
    {
    cout<<"\nPerson name   : "<<name;
    cout<<"\nPerson mob no : "<<mob<<endl;
    }
  }
  void dis()
  {
    cout<<"\nPerson details"<<endl;
    cout<<"\nPerson no     : "<<no;
    cout<<"\nPerson name   : "<<name;
    cout<<"\nPerson city   : "<<city;
    cout<<"\nPerson mob no : "<<mob<<endl;
  }

};
int main()
```

```
{
  char nme[10];
  int mno,i,no,ch;// mno=mobile no , no=total person no , ch=choice
  person p[20];
  do{
    cout<<"\n1.Accept person details\n2.Display person details\n3.To search the mobile number of a given person\n4.To search the Person details of a given mobile number\n5.Exit\nEnter your choice :- ";
  cin>>ch;
  switch(ch)
  {
  case 1:cout<<"Enter how many Person Details you want to enter: ";
              cin>>no;

      for(i=0;i<no;i++)
            {
                    p[i].acc();
```

```
            }
    break;
case 2:for(i=0;i<no;i++)
            p[i].dis();
    break;
case 3:cout<<"\nEnter person name search for mob no : ";
            cin>>nme;
    for(i=0;i<no;i++)
                p[i].acc(nme);
    break;
case 4:cout<<"\nEnter mob no search for person name : ";
                cin>>mno;
                for(i=0;i<no;i++)
                p[i].acc(mno);
    break;
}
}while(ch!=5);

return 0;
```

}

Slip 8 - A) Write a PHP Script to create class Shape and its sub-class Triangle, Square, Circle and display area of selected shape (use concept of inheritance).

Solution:

slip8-p1-q1.html

```
<HTML>
<BODY bgcolor=blue>
<FORM ACTION="slip8-p2-q1.php" METHOD="post">
<pre>
<h1>Circle:</h1>

    Radius: <input type=text name=a><BR>

<h1>
square:</h1>

    side: <input type=text name=b><BR>
<h1>
Rectangle:
```

</h1>

length: <input type=text name=c>

Breadth: <input type=text name=d>

<input type=submit value=submit> <input type="reset" value=cancel>

</pre>

</form>

</BODY>

</HTML>

slip8-p2-q1.php

<HTML>

<BODY>

<?PHP

$a=$_POST['a'];

$b=$_POST['b'];

$c=$_POST['c'];

$d=$_POST['d'];

```
interface one
{
    function area($c,$d);
}
class rect implements one
{
    function area($c,$d)
    {
        $area=$c*$d;
        echo"Area of rectangle.....:".$area;
        echo"<br><BR>";
    }
}
class square extends rect
{
    function area($b,$f)
    {
        $area=$b*$f;
```

```
            echo"Area of Square.....:".$area;
            echo"<BR><br>";
        }
}
class circle
{
        function area($a)
        {
            $area=3.14*$a*$a;
            echo"Area of circle.....:".$area;
            echo"<br><BR>";
        }
}
$o=new rect();
$o->area($c,$d);
$s=new square();
$f=$b;
$s->area($b,$f);
$c=new circle();
```

```
$c->area($a);
echo "<br>";
?>
<a href="a4b1.html">come back</a><br>
</BODY>
</html>
```

Slip 8 - B) Write an Ajax script to get player details from XML file when user select player name. Create XML file to store details of player (name, country, wickets and runs).

Solution:

slip8-p1-q2.html

```
<html>
<script type="text/javascript">
function display()
{
	name=f1.txt.value;
	var xmlhttp;
	if(window.XMLHttpRequest)
	{
		xmlhttp= new XMLHttpRequest();
```

```
        }
        else
        {
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
      xmlhttp.open("GET", "slip8-p1-q2.php?txt="+ name,false);
       xmlhttp.send();
       document.getElementById('result').innerHTML=
                xmlhttp.responseText;
}
</script>
<body>
        <form name="f1">
                Enter Employee name:
                <input type="text" name="txt"
onKeyUp="display()"/><br>
        </form>
         <div id='result'></div>
        </body>
```

```
</html>
```

slip8-p1-q2.php

```
<?php

$d=new DOMDocument();
$d->load("slip8-p1-q2.xml");

$run=$d->getElementsByTagName('runs');
$wic=$d->getElementsByTagName('wickets');
$name=$d->getElementsByTagName('player');

foreach($name as $n)
{
        if($a='')
        echo "<br>".$n->textContent;
       else "not";
}
?>
```

slip8-p1-q2.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<cricketinfo>
	<cricket>
		<player>abc</player>
		<runs>1000</runs>
		<wickets>50</wickets>
		<noofnotout>10</noofnotout>
	</cricket>

	<cricket>
		<player>def</player>
		<runs>100</runs>
		<wickets>40</wickets>
		<noofnotout>10</noofnotout>
	</cricket>

	<cricket>
```

```
        <player>pqr</player>
        <runs>1020</runs>
        <wickets>60</wickets>
        <noofnotout>10</noofnotout>
</cricket>

<cricket>
        <player>xyz</player>
        <runs>9000</runs>
        <wickets>90</wickets>
        <noofnotout>40</noofnotout>
</cricket>

<cricket>
        <player>lmn</player>
        <runs>170</runs>
        <wickets>80</wickets>
        <noofnotout>8</noofnotout>
</cricket>
```

</cricketinfo>

Slip 9 - A) Considerthe following C++ class class Person { char Name [20]; charA ddr[30]; float Salary; float tax_amount; public: // member functions }; Calculate tax amount by checking salary of a person For salary <=20000 tax rate=0 For salary >20000 || <=40000 tax rate= 5% of salary. For salary >40000 tax rate =10% of salary.

Solution:

```
#include<iostream.h>

#include<conio.h>

class person

{

char name[20];

char addr[20];
```

```
float sal,tax;

 public:

void get()

{

cout<<"Enter the name, address, salary : \n";

cin>>name>>addr>>sal;

}

void put()

{

cout<<"Person Information:\n";
```

```
cout<<"Name\tAddress\tSalary\tTax: \n";

cout<<"=========================================
=======\n";

cout<<name<<"\t"<<addr<<"\t"<<sal<<"\t"<<tax<<endl;

}

void cal_tax()

{

if(sal<=20000) //salary <=20000

{

tax=0;
```

```
}

else if(sal>=20000||sal<=40000)//salary >20000 11<
=40000   tax rate= 5% of salary.

{

tax=(sal*5)/100;

}

else if(sal >40000) //salary >40000  tax rate =10% of salary

{

tax=(sal*10)/100;

}
```

```
}

};

void main()

{

person p;

clrscr();

p.get();

p.cal_tax();

p.put();
```

```
getch();


}
```

Slip 9 - B) Create a C++ class Time with data members Hours, Minutes and Seconds. Write necessary member functions for the following: (Use Objects as arguments) i. To accept a time. ii. To display a time in format hh:mm:ss. iii. To find difference between two time and display it in format hh:mm:ss.

Solution:

```
#include<iostream.h>


#include<conio.h>


#include<iomanip.h>


class time


{
```

```
int h,m,s;

public:

void getdata();

void display();

time operator-(time t2);

};

void time::getdata()

{

cout<<"\nEnter hour,minutes and seconds\n";

cin>>h>>m>>s;
```

```
}

void time::display()

{

cout<<"\nTime is-> "<<setfill('0')<<setw(2)<<h;

cout<<":"<<setfill('0')<<setw(2)<<m;

cout<<":"<<setfill('0')<<setw(2)<<s<<endl;

}

time time::operator-(time t2)

{
```

```
time t;

t.h=h-t2.h;

t.m=m-t2.m;

t.s=s-t2.s;

return t;

}

void main()

{

clrscr();

time t1,t2,t3;
```

```
t1.getdata();

t1.display();

t2.getdata();

t2.display();

t3=t1-t2;

cout<<"\nTime1 - Time2:\n";

t3.display();

getch();
}
```

Slip 9 - A) Write a PHP program to create a Class Calculator which will accept two values from user and pass as an argument through parameterized constructor and do the

following task: a) Add b) Subtract c) Multiply them together or divide them on request.

Solution:

```
<?php
class Calculate
{
public $a;
public $b;
function __construct($a,$b){
$this->a=$a;
$this->b=$b;
}
public function add()
	{
	 $c=$this->a+$this->b;
echo"Addition = $c<br>";
	}
public function subtract()
	{
```

```
        $c=$this->a-$this->b;
echo"Subtract = $c<br>";
        }
public function multiply()
        {
        $c=$this->a*$this->b;
echo"Multiplication = $c<br>";
        }
public function div()
        {
        $c=$this->a/$this->b;
echo"Division = $c";
        }}
//$x=$_GET['a'];
//$y=$_GET['b'];
$calc=new Calculate(4,3);
$calc->add();
$calc->subtract();
$calc->multiply();
```

$calc->div();?>

Slip 9 - B) Consider the following entities and their relationships Movie(movie no,movie_name,release_year) Actor(actor no,name) Relationship between movie and actor is many-many with attribute rate in Rs. Create a RDB in 3 NF. With using three radio buttons (accept, insert, update) Write an AJAX script to accept actor name and display names of movies in which he has acted.

Solution:

html file -

<html>

<body>

<form action="slip_22.php" method="get">

<h3>Enter Actor Name : <input type=text name=nm> </h3>

<input type=radio name=a value=1>Display Movie Name

<h3>Enter movie no :<input type=text name=m_no>

<h3>Enter movie name :<input type=text name=m_nm>

<h3>Enter release year :<input type=text name=r_yr>

<h3>Enter actor no :<input type=text name=a_no>

<h3>Enter actor name :<input type=text name=a_nm>

<input type=radio name=a value=2>Insert New movie info

```
<input type=submit value=OK>
</form>
<div id="place"></div>
</body>
</html>
```

php file -

```
<?php
$r = $_GET['a'];
$con = mysql_connect("localhost","root","");
$d = mysql_select_db("bca_programs",$con);

if($r == 1)
{ $actor_name = $_GET['nm'];
$q = mysql_query("select m_name from movie,actor,movie_actor where movie.m_no=movie_actor.m_no and actor.a_no=movie_actor.a_no and a_name='$actor_name'");
echo "
```

```
Movie Name
";
while($row=mysql_fetch_array($q))
{
echo $row[0]."
";
}
}
else if($r == 2)
{ $m_no = $_GET['m_no'];
$m_name = $_GET['m_nm'];
$r_yr = $_GET['r_yr'];
$a_no = $_GET['a_no'];
$a_name = $_GET['a_nm'];
$q = mysql_query("insert into movie values($m_no,'$m_name',$r_yr)");
$q1 = mysql_query("insert into actor values($a_no,'$a_name')");
echo "Value Inserted";
}
```

```
mysql_close();
?>
```

Slip 10 - A) Write a C++ program to create a class Account with data members Acc_number, Acc_type and Balance. Write member functions to accept and display 'n' account details. (Use dynamic memoryallocation).

Solution:

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
 class Account
{
public:
int Acc_no,Balance;
char Acc_type[30];
public:
Account()  {  cout << "Constructor" << endl; }
~Account() { cout << "Destructor" << endl; }
void get_data()
{
```

```
cout<<"\n Enter Acc_no.:";
cin>>Acc_no;
cout<<"\n Enter Acc_type :";
cin>>Acc_type;
cout<<"\n Enter Balance :";
cin>>Balance;
}
void display_data()
{
cout<<"\t"<<Acc_no<<"\t"<<"\t"<<Acc_type<<"\t"<<Balance
;
}
};
int main()
{
 clrscr();
int num;
 Account* a = new Account[4];
delete [] a; // Delete array
```

```
 cout<<"\n How many records u want?: ";
cin>>num;
 for(int i=0;i<num;i++)
{
a[i].get_data();
}
for(i=0;i<num;i++)
{
a[i].display_data();
}
return 0;
}
```

Slip 10 - B) Create a C++ class City with data members City_code, City_name, population. Write necessary member functions for the following: i, Accept details of n cities ii. Display details of n cities in ascending order of population. iii. Display details of a particular city. (Use Array of object and to display city information use manipulators.)

Solution:

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

```
#include<conio.h>

#include<iomanip.h>

#include<stdlib.h>

#include<string.h>

 void searchcity();

 char n[10],c[10];

 class city

 {

public:
```

```
int population,city_code;

char name[40],e[10];

 void accept()

{

 cout<<"\n Enter name of city:";

cin>>name;

 cout<<"\n Enter name of city_code:";

cin>>city_code;

 cout<<"\n Enter the population of city:";

cin>>population;
```

```
}

void sort(city &r1,city &r2)

{

city rt;

if(r1.population>r2.population)

{

 rt=r1;

 r1=r2;

 r2=rt;
```

```
}

}

 void display()

{

cout<<"\n Name of City :"<<setw(15)<<name<<endl;

cout<<"\n Population :"<<setw(15)<<population<<endl;

cout<<"\n City Code:"<<setw(15)<<city_code<<endl;

}

 void searchcity()

{
```

```
if(strcmp(name,c)==0)

{

cout<<"\n name: "<<name<<"\n Population.: "<<population;

}

}

};

void main()

{

clrscr();
```

```
city t[30];

int num,ch,population;

char cont;

cout<<"\n 1.Accept & display ";

cout<<"\n 2.Ascending";

cout<<"\n 3.Search by city";

do

 {

cout<<"\n Enter your choice: ";

cin>>ch;
```

```
switch(ch)

{

case 1: cout<<"\n How many records you want to enter: ";

cin>>num;

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

{

t[i].accept();

}

 for(i=0;i<num;i++)
```

```
{

t[i].display();

}

 break;

 case 2:

 for(i=0;i<num;i++)

{

for(int j=i+1;j<num;j++)

t[i].sort(t[i],t[j]);

t[i].display();
```

```
}

 break;

 case 3: cout<<"\n Enter city name: ";

cin>>c;

for(i=0;i<num;i++)

{

t[i].searchcity();

}

break;
```

```
}

 cout<<"\n Do you want to continue: ";

cin>>cont;

}

 while(cont=='Y'||cont=='y');

getch();

}
```

Slip 10 - A) Write a PHP Script to demonstrate the concept of Introspection for examining object. (Using any 3 predefined functions).

Solution:

```
<?php
class Myclass
{
```

```
        public $a;
        public $b=305;
        public $c='Akshay';
        function Myclass()
    {
            //Myclass function
        }
        function myfun1()
    {
            //functin
    }
        function myfun2()
    {
            //functin
    }
}
        $class=get_declared_classes();
        foreach($class as $cname)
        {
```

```
        echo"$cname<br>";
    }
    echo"<br>Class Methods are : <br>";
    $m=get_class_methods('Myclass');
    foreach($m as $mname)
    {
        echo"$mname<br>";
    }
    $cp=get_class_vars('Myclass');
    echo"class variables are :<br>";
    foreach($cp as $cpname => $v)
    {
        echo"$cpname : $v <br>";
    }

?>
```

Slip 10 - B) Write a PHP script to perform the following stack related operations- insert, delete and display. (Use concept of self processing form)

Solution:

```
<html>
<body>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF']; ?>">
Select options:

<input type="radio" value="push" name="d1">Push</input>
<input type="radio" value="pop" name="d1">Pop</input>
<input type="radio" value="display" name="d1">Display</input>
<input type="submit">
</form>
<?php
echo "
Index ARRAY.....
";
$a = array(1,2,3,4,5,6,7);
print_r($a);
print"<br>";
if($_SERVER["REQUEST_METHOD"] == "POST")
```

```
{
$opt = $_POST['d1'];
if($opt == 'push')
{
array_push($a,11);
print_r($a);
}

else if($opt == 'pop')
{
array_pop($a);
print_r($a);
}
else if($opt == 'display')
{
//array_pop($a);
print_r($a);
}
```

```
}
?>
```

Slip 11 - A) Create a C++ class MyArray, which contains single dimensional integer array of a given size. Write a member function to display sum of given array elements. (Use Dynamic Constructor and Destructor)

Solution:

```
#include<stdio.h>

#include<conio.h>

#include<iostream.h>

class MyArray

{

int size,*ptr,*p;

int sum=0;
```

```
public:

MyArray(int no)

{

size=no;

ptr=new int[size];

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

{

cout<<"enter elementS";

cin>>ptr[i];
```

```
sum = sum + ptr[i];

}

}

 void display()

{

cout<<"elements are";

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

{

cout<<ptr[i]<<"\t";

}
```

```
cout<<"\n\nSum of all elements is: "<<sum;

}

 Void sum()

{

        int arr[10];

        int i;

        int sum=0;

        cout<<"Enter 10 array elements: ";

        for(i=0; i<10; i++)
```

```
{

        cin>>arr[i];

        sum = sum + arr[i];

}

cout<<"\nThe array elements are: \n";

for(i=0; i<10; i++)

{

        cout<<arr[i]<<"  ";

}

cout<<"\n\nSum of all elements is: "<<sum;
```

```
 }

  ~MyArray()

{delete ptr;

}

 };

 void main()

{

int n;

clrscr();
```

```
cout<<"enter size";

cin>>n;

MyArray d(n);

d.display();

d.sum();

getch();

}
```

Slip 11 - B) Create a base class Student with data members Roll_No, Name. Derives two classes from it, class Theory with data members M1, M2, M3, M4 and class Practical with data members P1, P2. Class Result(Total_Marks, Percentage, Grade) inherits both Theory and Practical classes. (Use concept of Virtual Base Class and protected access specifiers) Write a C++ menudriven programto perform the following functions: i. Accept Student

Information ii. Display Student Information iii. Calculate Total_marks, Percentage and Grade.

Solution:

```
#include<conio.h>

#include<string.h>

#include<conio.h>

#include<iostream.h>

class student

{

protected:

int  rno;
```

```
char name[20];

public:

void getdetails();

};

 class Theory:public virtual student

{

protected:

int mark1,mark2,mark3,mark4;

public:

void getmarks();
```

```
};

 class Practical :virtual public student

{

protected:

int p1,p2;

public:

void getpractical();

};

 class result :public Theory,public Practical
```

```
{

int total_marks;

float per;

char grade[10];

public:

void calc();

void sort(result& ,result&);

void display();

};

 void student::getdetails()
```

```
{

cout<<"\n enter roll no and name  :";

cin>>rno>>name;

}

 void Theory::getmarks()

{

cout<<"\n enter marks of four subject :";

cin >>mark1>>mark2>>mark3>>mark4;

}
```

```
 void Practical::getpractical()

{

cout<<"\n enter  Practical Details  :";

cin>>p1>>p2;

}

 void result::calc()

{

int i;

total_marks=mark1+mark2+mark3+mark4+p1+p2;

per=total_marks/(float)6;
```

```
if(per<50)

strcpy(grade,"C");

else

if(per<60)

strcpy(grade,"B");

else

if(per<75)

strcpy(grade,"A");

else
```

```
strcpy(grade,"A+");

cout<<"\n calculation complete\n";

}

 void result::sort(result &r1,result &r2)

{

result rt;

if(r1.total_marks>r2.total_marks)

{

rt=r1;

r1=r2;
```

```
r2=rt;

}

}

 void result::display()

{

cout<<"\n roll no="<<rno<<"\n name ="<<name;

cout<<"\n mark1="<<mark1<<"\n mark2 ="<<mark2<<"\n mark3="<<mark3<<"\n mark4="<<mark4;

cout<<"\n Practical P1="<<p1<<"\n Practical P2="<<p2<<"\n percentage ="<<per<<"\n grade ="<<grade;

}
```

```
 int main()

{

int n,i,ch,j;

result r[20];

clrscr();

do

{

cout<<"\n MENU \n";

cout<<"\n 1.build master table \n 2. calculate total & grade
\n";
```

```
cout<<"\n 3.display result in asscending order \n enter your choice :  ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n how many student :";

cin>>n;

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

{
```

```
cout<<"enter student detailse \n";

r[i].getdetails();

r[i].getmarks();

r[i].getpractical();

}

break;

 case 2:

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

r[i].calc();
```

```
break;

 case 3:

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

{

for(j=i+1;j<n;j++)

r[i].sort(r[i],r[j]);

r[i].display();

}

break;

 }
```

```
}while(ch<=3);


getch();


return 0;


}
```

Slip 11 - B) Write an Ajax code to print the content of "Myfile.dat" on clicking on fetch Button. The data fetches from the server using Ajax Technique.

Solution:

Fetch.php

```
<html>
<head>
<script language="javascript">
var req=false;
if(window.XMLHttpRequest)
{
req=new XMLHttpRequest();
```

```
}
else if(window.ActiveXObject)
{
req=new ActiveXObject("Microsoft.XMLHttp");
}
function fetchdata(datasource,divID)
{
if(req)
{
req.open("GET",datasource);
req.onreadystatechange=function()
{
if(req.readyState==4 && req.status==200)
{
document.getElementById(divID).innerHTML=req.responseT
ext;
}
}
req.send(null);
```

```
}
}
</script>
</head>
<body>
<form>
<input type=button value="Fetch Message" onClick="fetchdata('myfile.txt','myDiv')">
</form>
<div id="myDiv">data will be here
</body>
</div>
</html>
```

Slip 12 - A) Write a C++ program to create a class Date with data members day, month, and year. Use default and parameterized constructor to initialize date and display date in dd-Mon-yyyy format. (Example: Input: 04-01-2021 Output: 04-Jan-2021).

Solution:

```
#include<iostream>
```

```
#include<conio.h>

using namespace std;
class date
{
 int dd,mm,yy;
 public:
 date(int d,int m,int y)
 {
 dd=d;
 mm=m;
 yy=y;
 }
 void display()
 {
 cout<<"\nGiven date is\t";
 cout<<dd<<"-"<<mm<<"-"<<yy;
 cout<<"\nAfter formating date is\t";
 switch(mm)
```

```
{
case 1:
cout<<"\n"<<dd<<"-Jan-"<<yy;
break;
case 2:
cout<<"\n"<<dd<<"-Feb-"<<yy;
break;
case 3:
cout<<"\n"<<dd<<"-Mar-"<<yy;
break;
case 4:
cout<<"\n"<<dd<<"-Apr-"<<yy;
break;
case 5:
cout<<"\n"<<dd<<"-May-"<<yy;
break;
case 6:
cout<<"\n"<<dd<<"-Jun-"<<yy;
break;
```

```
case 7:
cout<<"\n"<<dd<<"-Jul-"<<yy;
break;
case 8:
cout<<"\n"<<dd<<"-Aug-"<<yy;
break;
case 9:
cout<<"\n"<<dd<<"-Sep-"<<yy;
break;
case 10:
cout<<"\n"<<dd<<"-Oct-"<<yy;
break;
case 11:
cout<<"\n"<<dd<<"-Nov-"<<yy;
break;
case 12:
cout<<"\n"<<dd<<"-Dec-"<<yy;
break;
default:
```

```
 cout<<"\nInvalid month";


 }
 }
};
int main()
{
 int m,dt,y;
 cout<<"\n Enter date : ";
 cin>>dt;
 cout<<"\n Enter month : ";
 cin>>m;
 cout<<"\n Enter year : ";
 cin>>y;
 date d(dt,m,y);
 d.display();
 return 0;
}
```

Slip 12 - B) Create a C++ class Weight with data members kilogram, gram. Write a C++ program using operator overloading to perform following functions: i. To accept weight. ii. To display weight in kilogram and gram format. iii. Overload += operator to add two weights. iv. Overload == operator to check equality of two weights.

Solution:

```
#include<iostream.h>

#include<conio.h>

class Weight

{

int kilogram,gram;

public:

void getdata()
```

```
{

cout<<"\n\nEnter the kilogram:\t";

cin>>kilogram;

cout<<"\nEnter the gram:\t";

cin>>gram;

}

void display()

{

cout<<"\nAddition of two distance:\t";
```

```
cout<<kilogram<<"."<<gram;

}

void display2()

{

cout<<kilogram<<"."<<gram;

}

Weight operator+=(Weight &d)

{

Weight t;

t.kilogram=d.kilogram+kilogram;
```

```
t.gram=d.gram+gram;

return t;

}

int operator==(Weight &d)

{

if(kilogram==d.kilogram  || gram==d.gram)

{

return 1;

}
```

```
else

{

return 0;

}

}

};

void main()

{

Weight c1,c2,c3,c4,c5;

clrscr();
```

```
c1.getdata();

c2.getdata();

c3=c1+=c2;

c3.display();

c4.getdata();

c5.getdata();

if(c4==c5)

{

cout<<"\n";
```

```
c4.display2();

c5.display2();

cout<<"\t Both are same\t";

}

else

{

cout<<"\n";

c5.display2();

c4.display2();

cout<<"\t Both are not same\t";
```

```
}

getch();

}
```

Slip 12 - A) Write a PHP program to convert temperature Fahrenheit to Celsius using sticky form.

Solution:

```
<html>
<body>
<?php
$fahr=$_POST['fahrenheit']
<form action="<?php echo $_SERVER['PHP_SELF']?>"method="POST">
fahrenheit temperature:
<input type="text" name="fahrenheit" value=if(isset(=$_POST['fahrenheit'])){echo $fahr}?>>
<input type ="submit">
</form>
```

```
<?php
$Celsius=($fahr-32)*5/9;
printf("%f:%f,$fahr,$Celsius);
?>
</body>
</html>
```

Slip 12 - B) Write an AJAX script to read contact.dat file and print the content of a file in a tabular form when the user clicks on print button. Contact.dat file contains srno, name, residence number, mobile number (Enter at least 3 records in contact.datfile)

Solution:

HTML file :

```
<html>
<head>
<style>
span
{
        font-size: 25px;
}
```

```
table
{
        color: blueviolet; ;
}
</style>

<script type="text/javascript" >
        function print()
        {
                var ob=false;
                ob=new XMLHttpRequest();

                 ob.open("GET","slip14_Q2.php?");//emailid="
+eid);
                ob.send();

                ob.onreadystatechange=function()
                {
                        if(ob.readyState==4 && ob.status==200)
```

```
                        {
                                document.getElementById("i"
).innerHTML=ob.responseText;
                        }
                }
        }
</script>
</head>

<body>
<center>
<h3>Display the contents of a contact.dat file </h3>
<br><input  type="button"  value=Print onclick="print()" >
<span id="i"></span>
</center>
</body>
</html>
```

PHP file :

```
<?php
        $fp = fopen('contact.dat','r');
        echo "<table border=1>";
                                                    echo "<tr><th>Sr. No.</th><th>Name</th><th>Residence No.</th><th>Mob. no.</th><th>Relation</th></tr>";

while($row =  fscanf($fp,"%s %s %s %s %s"))
        {
                echo "<tr>";
                foreach($row as $r)
                {
                        echo "<td>$r</td>";
                }
                echo "</tr>";
        }
                echo "</table>";
        fclose($fp);
```

?