Set 3 (April 3)

Slip 1 a) a.Write C++ program to check maximum and minimum of two integer numbers (Use inline function and conditional operators)

```
#include<iostream>
#include<conio.h>
using namespace std;
class max1
{
public:

inline int maximum(int a,int b)
{
return a>b?a:b;
}
  inline int minimum(int a,int b)
{
return a<b?a:b;
}
};
int main()
{
int a,b;
max1 m;
cout<<"\n Enter two number:";
cin>>a>>b;

cout<<"\nNumber 1st: " <<a<<endl;
cout<<"\nNumber 2nd: "<<b<<endl;
 cout<<"\nMaximum number is: "<<m.maximum(a,b)<<endl;
 cout<<"\nMinimum is: "<<m.minimum(a,b)<<endl;
}
```

Slip 2 a.)Write a C++ program to calculate Volume of cone,sphere and cylinder by using function overloading.

```
 #include<iostream>
using namespace std;
float vol(int,int);//cylinder
float vol(float);//sphere
int vol(float,int);//cone

int main()
{
int radius,height,radius2,height2;
float radius1;

cout<<"Enter radius and height of a cylinder:";
```

```
cin>>radius>>height;

cout<<"Enter radius of sphere: ";
cin>>radius1;

cout<<"Enter radius and height of a cone:";
cin>>radius2>>height2;

cout<<"\n Volume of cylinder is "<<vol(radius,height);//cylinder
cout<<"\n Volume of sphere is "<<vol(radius1);//sphere
cout<<"\n Volume of cone is "<<vol(radius2,height2);//cone

return 0;
}

float vol(int radius,int height)//cylinder
{
return(3.14*radius*radius*height);
}

float vol(float radius1)//sphere
{
return((4*3.14*radius1*radius1*radius1)/3);
}

int vol(float radius2,int height2)//cone
{
return(0.33*3.14*radius2*radius2*height2);
}
```

Slip 3 a) **Write a C++ program to create a class which contains two data members. Write member functions to accept display and swap two entered numbers using call by reference.**

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

void swap(int &x,int &y)
{
        int temp;
        temp = x;
        x = y;
        y = temp;
        return;
}
int main()
{
        int a = 100;
        int b = 200;
```

```
    cout<<"Before swap ,value of a :" <<a<<endl;
    cout<<"Before swap ,value of b :" <<b<<endl;
    swap(a,b);
    cout<<"After swap ,value of a :" <<a<<endl;
    cout<<"After swap ,value of b :" <<b<<endl;
    return 0;
}
```

Slip 4 a)Write a C++ program to create a class Worker with data members as Worker_Name, No_of_Hours_worked, Pay_Rate. Write necessary member functions to calculate and display the salary of worker. (Use default value for Pay_Rate)

```
#include<iostream>
using namespace std;
class worker
{
     char name[10];
     int hr;
     public:
     void accept()
     {
        cout<<"enter name";
        cin>>name;
        cout<<"enter hours";
         cin>>hr;
      }
      void calculate(int rate=20)
      {
         cout<<"salary of worker is Rs."<<(hr*10)*rate;
       }
     };

int main()
{
    worker w;
    w.accept();
    w.calculate();

}
```

Slip 6a) /*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 .*/

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

```
{
friend class Rectangle;      // declaring Rectangle as friend class
int side;
public:
Square ( int s )
{
side = s;
}
};

class Rectangle
{
int length;
int breadth;
public:
int getArea()
{
return length * breadth;
}
void shape( Square a )
{
length = a.side;
breadth = a.side;
}
};

int main()
{
Square square(5);
Rectangle rectangle;
rectangle.shape(square);
cout << " Area of rectangle is   "<<rectangle.getArea();
return 0;
}
```

Slip 8a) /*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().*/

```
#include<iostream>
using namespace std;
class Number
{
static int cnt;
public:
static void display()
{
```

```
cout<<"\n Number of time call show: "<<cnt;
cnt++;
}
};
int Number::cnt;
int main()
{
Number s1,s2,s3,s4;
s1.display();
s2.display();
s3.display();
s4.display();
return 0;
}
```

Slip 9 a) /*Consider the following C++ class

```
class Person
{
char Name [20];
char Add r [30];
float Salary;
float  tax_amount;
public:
II 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.*/

```
#include<iostream>
using namespace std;
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;
}
}
};
int main()
{
person p;
p.get();
p.cal_tax();
p.put();
}
```

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 memory allocation)*/

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
 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()
{

 int num,i;
 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;
 }
```