OPPs Sem-II

Assignment-IV

6/22/2021

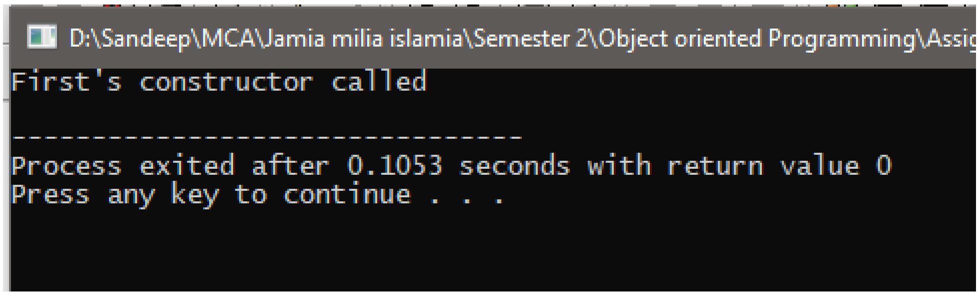
Sandeep Bhatt

Q1 : Write a program to implement the usage of static data members and static member functions of a class.

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

class first
{
	int x;
public:
	First() { cout << "First's constructor called " << endl; }
};

class second
{
	static first a;
public:
	B() { cout << "second's constructor called " << endl; }
	static first getfirst() { return first; }
};

first second::first; // definition of first

int main()
{
	// static member 'first' is accessed without any object of second
	first a = second::getfirst();
```

```
        return 0;

}
```

Output:



```
D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assig
First's constructor called

--------------------------------
Process exited after 0.1053 seconds with return value 0
Press any key to continue . . .
```

2. Write a program to generate results for 10 students using two classesStudent and Exam. Student class contains- stud_roll, stud_name, course_name, dob as data members and getrec(), modifyrec(), printrec() as member functions. Exam class contains- exam_name, paper_name,paper_code, marks_obtained, total marks, grades as data members and getmarks(), modifymarks(),printresult() as member functions.

```
#include<iostream>

#include<stdio.h>


using namespace std;


class student{

  char stud_name, course_name;

  int stud_rollno,dob;

  public:

  void getrec(int);

  void modifyrec();

  char printrec(){ return stud_name; }
```

```
};


class exam:public student{
   int  paper_code,marks_obtained,total_marks;
   char exam_name, paper_name,grades;
   public:
   void getmarks();
   void modifymarks();
   void printresult();
};


void student::getrec(int i)
{
  char buffer[255];
  bool isvalid;
  fgets(buffer, sizeof(buffer), stdin); // clearing standard input stream
    do{
      cout << endl << "Enter name of student no " << i << " : ";
      getline(cin, name);
      isvalid = isValidName(name);
      if(!isvalid) cout << "Invalid Name...! Try Again." << endl;
     }while(!isvalid);


    do{
      cout << "Enter " << stud_name << "'s rollno : ";
      cin >> rollno;
      if(rollno < 1) cout << "Invalid input...! Try again" << endl;
    }while(rollno < 1);


      fgets(buffer, sizeof(buffer), stdin); // clearing standard input stream
    do{
       cout << "Enter " << stud_name << "'s course : ";
       getline(cin, course);
       isvalid = isValidName(course);
       if(!isvalid) cout << "Invalid Name...! Try Again." << endl;
     }while(!isvalid);


}

void student::printrec()
{
  cout << endl << "Name = " << stud_name
       << endl << "Roll no = " << rollno
       << endl << "Course = " << course;
}

void exam::getmarks()
{
  do{
     cout << "Enter " << getname() << "'s marks in History ( 0 <= M <= 100 ): ";
     cin >> marks1;
     if(marks1 < 0 || marks1 > 100) cout << "Invalid input...! Try again" << endl;
    }while(marks1 < 0 || marks1 > 100);
  do{
     cout << "Enter " << getname() << "'s marks in Political Science ( 0 <= M <= 100 ): ";
     cin >> marks2;
     if(marks2 < 0 || marks2 > 100) cout << "Invalid input...! Try again" << endl;
    }while(marks2 < 0 || marks2 > 100);

  do{
     cout << "Enter " << getname() << "'s marks in Urdu  ( 0 <= M <= 100 ): ";
     cin >> marks3;
     if(marks3 < 0 || marks3 > 100) cout << "Invalid input...! Try again" << endl;
    }while(marks3 < 0 || marks3 > 100);
}

void exam::printresult()
{
    cout << endl << "OPPS Marks : " << marks1
         << endl << "DBMS Marks : " << marks2
         << endl << "Advance data structure marks : " << marks3;
}

int main()
{
    int n;

    do{
       cout << endl << "Enter no. of students : ";
       cin >> n;
```

```
      if( n < 1) cout << "Invalid input...! Try again." << endl;
    }while(n < 1);


    exam s[n];
    for( int i = 0 ; i < n ; i++ )
    {
       s[i].input_student(i + 1 );
       s[i].input_marks();
       cout << endl;
    }
    cout << endl << "Student Details :";


    for( int i = 0 ; i < n ; i++ )
    {
       s[i].display_student();
       s[i].display_marks();
       cout << endl;
    }
    return 0;
}
```

3. Write a program to implement the member functions of a Class Shape having the same name, calculate_area() for calculating the area of a Triangle, Rectangle and Circle using the concept of Function overloading.

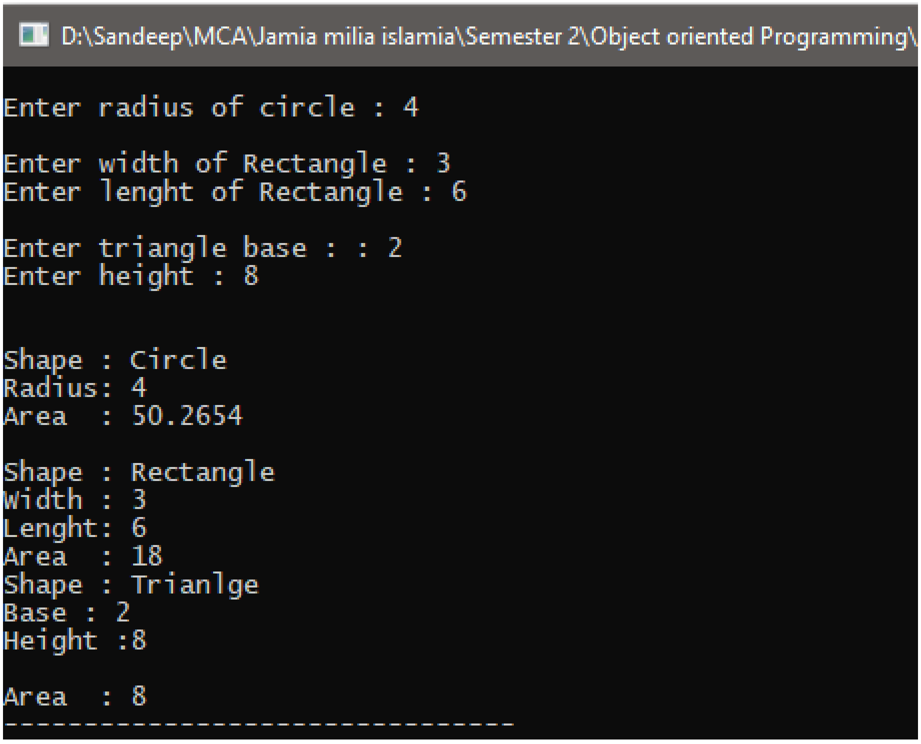
```
#include<iostream>


using namespace std;
```

```
class shape{

public:

 virtual void area(){};

 virtual void display(){};

};


class circle:public shape{

 double radius;

public:

 void getdata();

 void display();

 void area();

};


class rectangle:public shape{

  double width, length;

 public:

  void getdata();

  void display();

  void area();

};


class triangle:public shape{

 double base, height;

public:

 void getdata();

 void display();
 void area();

};

void circle::getdata()
{
    cout << endl << "Enter radius of circle : ";
    cin >> radius;
}
void circle::display()
{
    cout << endl << endl << "Shape : Circle" << endl << "Radius: " << radius;
}
void circle::area()
{
 cout << endl << "Area  : " << 3.14159 * radius * radius;
}
void rectangle::getdata()
{
    cout << endl << "Enter width of Rectangle : ";
    cin >> width;
    cout << "Enter lenght of Rectangle : ";
    cin >> length;
}
void rectangle::display()
{
    cout << endl << endl << "Shape : Rectangle" << endl << "Width : " << width << endl << "Lenght: " << length;
```

```
}
void rectangle::area()
{
 cout << endl << "Area  : " << width * length;
}


void triangle::getdata()
{
   cout << endl << "Enter triangle base : : ";
   cin >> base;
   cout << "Enter height : ";
   cin >> height;
}
void triangle::display()
{
   cout << endl << "Shape : Trianlge" << endl << "Base : " << base << endl << "Height :"<< height
<<endl;
}
void triangle::area()
{
 cout << endl << "Area  : " << ((base *height) / 2);
}


int main()
{
   shape* p;
   circle c;
   rectangle r;
```

```
    triangle t;


    p = &c;
    c.getdata();
    p = &r;
    r.getdata();
    p = &t;
    t.getdata();


    p = &c;
    p->display();
    p->area();


    p = &r;
    p->display();
    p->area();


    p = &t;
    p->display();
    p->area();


    return 0;
}
```

OUTPUT:



D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\

```
Enter radius of circle : 4

Enter width of Rectangle : 3
Enter lenght of Rectangle : 6

Enter triangle base : : 2
Enter height : 8


Shape : Circle
Radius: 4
Area  : 50.2654

Shape : Rectangle
Width : 3
Lenght: 6
Area  : 18
Shape : Trianlge
Base : 2
Height :8

Area  : 8
--------------------------------
```

4. Write a program with Student as abstract class and create derive classes- Engineering, Medicine and Science from base class Student. Create the objects of the derived classes and process them and access them using array of pointers of type base class Student. Include the relevant data members, constructors/destructors and member functions in each of the above classes.

```
#include<iostream>

#include<stdio.h>

#include<iomanip>



using namespace std;



class student{

  protected:

  string name;
```

```
    int rollno;

    int age;

    public:

    virtual void getdata(int)=0;

    virtual void putdata(int)=0;

};


class engineering:public student{

    public:

    void getdata(int);

    void putdata(int);

};

class medicine:public student{

    public:

    void getdata(int);

    void putdata(int);

};

class science:public student{

    public:

    void getdata(int);

    void putdata(int);

};


void engineering::getdata(int i)

{

    bool isValid;

    char buffer[255];
```

```
   fgets(buffer, sizeof(buffer), stdin);


   do{
     cout << endl << "Enter name of student " << i << " : ";
     getline(cin, name);
     isValid = isValidName(name);
     if(!isValid) cout << "Invalid Name...! Try Again." << endl;
    }while(!isValid);


   do{
     cout << "Enter " << name << "'s rollno : ";
     cin >> rollno;
     if(rollno < 1) cout << "Invalid input...! Try Again." << endl << endl;
    }while(rollno < 1);


   do{
     cout << "Enter " << name << "'s age : ";
     cin >> age;
     if(age < 1) cout << "Invalid input...! Try Again." << endl << endl;
    }while(age < 1);
}
void engineering::putdata(int i)
{
   cout << endl << endl << setw(2) << setfill('0')  << i << ") Name   : " << name << endl << "    Roll no: " << rollno << endl <<"    Age    : " << age;
}


void medicine::getdata(int i)
```

```
{
   bool isValid;
   char buffer[255];
   fgets(buffer, sizeof(buffer), stdin);

   do{
      cout << endl << "Enter name of student " << i << " : ";
      getline(cin, name);
      isValid = isValidName(name);
      if(!isValid) cout << "Invalid Name...! Try Again." << endl;
     }while(!isValid);

   do{
      cout << "Enter " << name << "'s rollno : ";
      cin >> rollno;
      if(rollno < 1) cout << "Invalid input...! Try Again." << endl << endl;
     }while(rollno < 1);

   do{
      cout << "Enter " << name << "'s age : ";
      cin >> age;
      if(age < 1) cout << "Invalid input...! Try Again." << endl << endl;
     }while(age < 1);
}
void medicine::putdata(int i)
{
   cout << endl << endl << setw(2) << setfill('0')  << i << ") Name   : " << name << endl << "    Roll no: "
<< rollno << endl <<"    Age    : " << age;
```

```
}



void science::getdata(int i)

{

    bool isValid;

    char buffer[255];

    fgets(buffer, sizeof(buffer), stdin);


    do{

       cout << endl << "Enter name of student " << i << " : ";

       getline(cin, name);

       isValid = isValidName(name);

       if(!isValid) cout << "Invalid Name...! Try Again." << endl;

      }while(!isValid);


    do{

       cout << "Enter " << name << "'s rollno : ";

       cin >> rollno;

       if(rollno < 1) cout << "Invalid input...! Try Again." << endl << endl;

      }while(rollno < 1);


    do{

       cout << "Enter " << name << "'s age : ";

       cin >> age;

       if(age < 1) cout << "Invalid input...! Try Again." << endl << endl;

      }while(age < 1);
```

```
}
void science::putdata(int i)
{
   cout << endl << endl << setw(2) << setfill('0')  << i << ") Name   : " << name << endl << "    Roll no: "
<< rollno << endl <<"    Age    : " << age;
}

int main()
{
   student *s;
   int nE, nM, nS, i;

   do{
     cout << "Enter no of Engineering students: ";
     cin >> nE;
     if(nE < 1) cout << "Invalid Input...! Try Again." << endl << endl;
    }while(nE < 1);

   do{
     cout << "Enter no of Medicine students   : ";
     cin >> nM;
     if(nM < 1) cout << "Invalid Input...! Try Again." << endl << endl;
    }while(nM < 1);

   do{
     cout << "Enter no of Science students    : ";
     cin >> nS;
     if(nS < 1) cout << "Invalid Input...! Try Again." << endl << endl;
```

```
  }while(nS < 1);

engineering sE[nE];  // engineering student
medicine sM[nM];     // medicine student
science sS[nS];      // science student

cout << endl << "You have to give details of " << nE << " Engineering Students :";
for(i = 0 ; i < nE ; i++)
{
  s = &sE[i];
  s->getdata(i + 1);
}

cout << endl << endl << "You have to give details of " << nM << " Medicine Students :";
for(i = 0 ; i < nM ; i++)
{
  s = &sM[i];
  s->getdata(i + 1);
}

cout << endl << endl << "You have to give details of " << nS << " Science Students :";
for(i = 0 ; i < nS ; i++)
{
  s = &sS[i];
  s->getdata(i + 1);
}
```

```
    for(i = 0, cout << endl ; i < 32 ; i++, cout << "*");

    cout << endl << "* Engineering Students Details *" << endl;

    for(i = 0 ; i < 32 ; i++, cout << "*");

    for(i = 0 ; i < nE ; i++) s = &sE[i], s->putdata(i + 1);




    for(i = 0, cout << endl ; i < 32 ; i++, cout << "*");

    cout << endl << "* Medicine Students Details    *";

    for(i = 0, cout << endl ; i < 32 ; i++, cout << "*");

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

      s = &sM[i], s->putdata(i + 1);


    for(i = 0, cout << endl ; i < 32 ; i++, cout << "*");

    cout << endl << "* Science Students Details     *";

    for(i = 0, cout << endl ; i < 32 ; i++, cout << "*");

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

     s = &sS[i], s->putdata(i + 1);


    return 0;

}
```

OUTPUT: