OPPs Sem-II

Assignment-III

5/30/2021

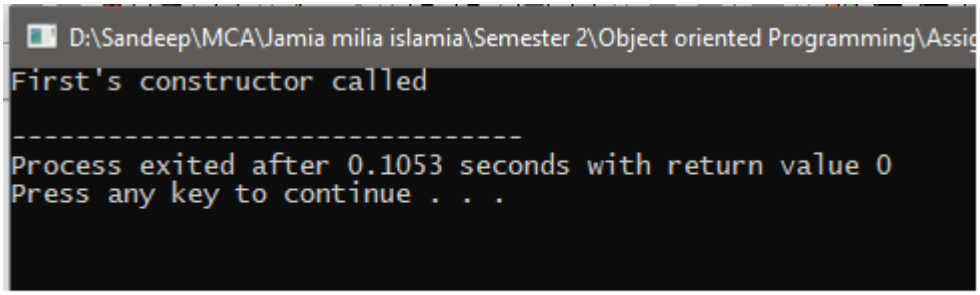
Sandeep Bhatt

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:



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;
```

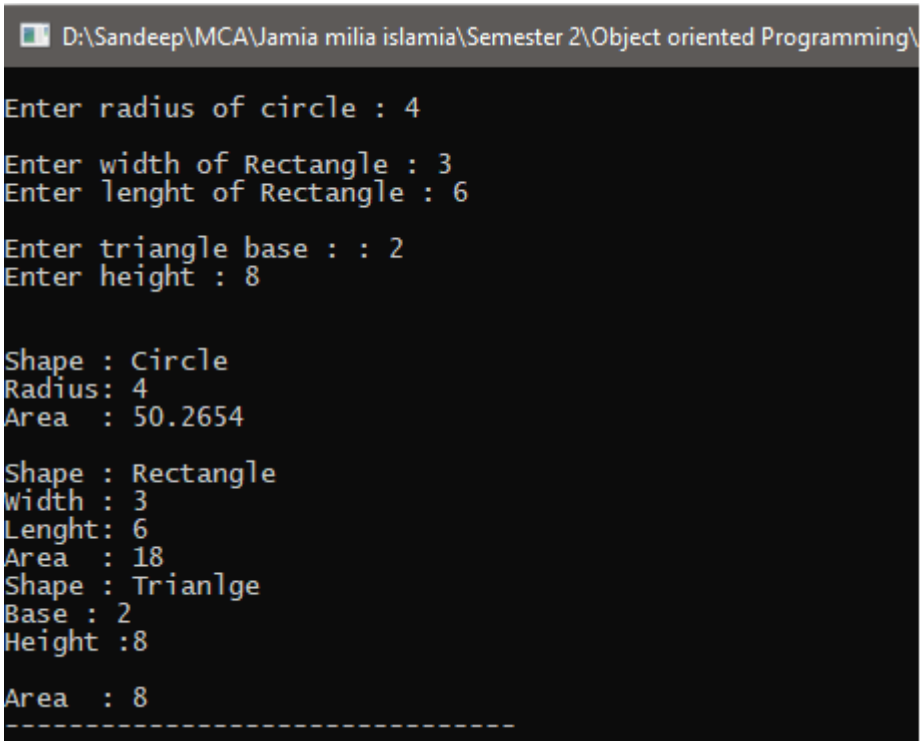
```
class shape{
public:
 virtual void area(){};
 virtual void display(){};
};

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:



4. Write a program to convert a distance entered in Feet and Inches to Meters using class to basic data type conversion.

```
#include<iostream>

#include<iomanip>


using namespace std;


class dist{

  int inches;

  int feets;


public:

  void getdata();

  float getInches(){ return inches; };
```

```
  float getFeets(){ return feets; }
  operator float()
  {
     float totalFeets;
     totalFeets = feets + inches*1/12;

     return totalFeets*1/3.28;
  }
};

void dist::getdata()
{
  cout << endl << "Enter distance in  inches : ";
  cin >> inches;
  cout << "Enter distance in feets : ";
  cin >> feets;
}

int main()
{
  dist d;
  float m;

  d.getdata();
  m = d;
  cout << endl << "Meters(" << d.getFeets() << " F + " << d.getInches() << " I) = " << m;
```

```
  return 0;

}
```

OUTPUT: