# Object-Oriented Application Development

**Practical 4**

**Part A**

1. Examine and run the following programs

(a) Program creates an array of double.

|  |
| --- |
| using System;  public class ArrayExample1  {  public static void Main()  {  double[] results;  results = new double[4];  results[0] = 8.4;  results[1] = 2.5;  results[2] = 6.1;  results[3] = 5.7;  Console.WriteLine("Length of array is " + results.Length);  Console.WriteLine("Values in the array are: ");  for (int i = 0; i < results.Length; i++)  {  // display to 1 decimal place  Console.WriteLine("{0:F1} ", results[i]);  }  }  } |

1. What will be the value of results.Length when the program executes?

4

1. What will be output of the program?
2. Rewrite the program to initialize the array using an array initializer as follows:

{8.4, 2.5, 6.1, 5.7}

and to print the array values using a *foreach* statement instead of a *for* statement.

foreach (double a in results)

Console.WriteLine(a);

(b) Program uses foreach loop to access values of array

|  |
| --- |
| using System;  public class ArrayExample3  {  public static void Main()  {  string[] colors = { "red", "green", "blue" };  foreach (string aColor in colors)  Console.WriteLine(aColor);  }  } |

(c) Main() method passes array as parameter to methods

|  |
| --- |
| using System;  public class ArrayExample4  {  public static void Main()  {  double[] temperature = new double[5];  InputValues(temperature);  DisplayValues(temperature);  }  // static method with array parameter  public static void InputValues(double[] temp)  {  string inValue;  for (int i = 0; i < temp.Length; i++)  {  Console.Write("Enter a temperature: ");  inValue = Console.ReadLine();  temp[i] = Convert.ToDouble(inValue);  }  }  // static method with array parameter  public static void DisplayValues(double[] temp)  {  Console.WriteLine("Temperatures");  foreach (double reading in temp)  {  Console.WriteLine(reading);  }  }  } |

(d) Main() method receives array as return value from method.

|  |
| --- |
| using System;  public class ArrayExample5  {  public static void Main()  {  double[] temperature;  temperature = InputReadings();  DisplayValues(temperature);  }  // static method which returns an array  public static double [] InputReadings()  {  double [] temp = new double[5];  string inValue;  for (int i = 0; i < temp.Length; i++)  {  Console.Write("Enter a temperature: ");  inValue = Console.ReadLine();  temp[i] = Convert.ToDouble(inValue);  }  return temp;  }  // static method with array parameter  public static void DisplayValues(double[] temp)  {  Console.WriteLine("Temperatures");  foreach (double reading in temp)  {  Console.WriteLine(reading);  }  }  } |

1. The following program contains a Player class and a PlayerTest class. The Player class has the following:

* Instance variables and associated properties representing the name (*string*) and number of games played (*integer*)
* An instance variable representing the points scored for each game (*array of integers*).
* A constructor with two parameters representing name and number of games played.
* An instance method with an array parameter representing the scores for each game played. The method copies the array parameter values to the instance variable representing the points scored.
* Another instance method that computes and returns the average score.

Examine and run the program.

|  |
| --- |
| public class Player  {  private string name;  public string Name  {  get { return name; }  set { name = value; }  }  private int numberOfGames;  public int NumberOfGames  {  get { return numberOfGames; }  set { numberOfGames = value; }  }  // array instance variable to store points scored for each game played  private int[] pointsScored;  // constructor with 2 parameters  public Player(string aName, int gamesPlayed)  {  Name = aName;  NumberOfGames = gamesPlayed;  // create array for pointsScored – size depends on number of games played  pointsScored = new int[NumberOfGames];  }  // instance method with array parameter  public void FillScores(int [] scores)  {  // copy values from array parameter to pointsScored  for (int i = 0; i < scores.Length; i++)  pointsScored[i] = scores[i];  }  // instance method to compute and return average score  public double GetAverage()  {  double total = 0.0;  double average;  foreach (int aScore in pointsScored)  {  total += aScore;  }  average = total / numberOfGames;  return average;  }  } |

|  |
| --- |
| using System;  public class PlayerTest  {  public static void Main()  {  Player player = new Player("Jordan", 3);  int[] scores = { 5, 12, 3 };  player.FillScores(scores);  Console.WriteLine("Average is {0:F0}", player.GetAverage());  }  } |

|  |
| --- |
| Player |
| -name :string  -numberOfGames:int  -pointScored: int{} |
| +Player (aName:string, gamesPlayed:int)  +FillScores(scores:int[]):void  +GeAverage():double |

3. The following program contains a RainfallForWeek1 class and a RainfallForWeekTest1 class. The RainfallForWeek1 class contains the following:

* An instance variable which is an array to store rainfall readings for 7 days of a week.
* A constructor with no parameters
* An instance method with two parameters representing a day and the rainfall reading for the day. The method stores the reading into an element of the array.
* An instance method with one parameter representing a day. The method that returns the rainfall reading from the array for that day.

Examine and run the program.

|  |
| --- |
| public class RainfallForWeek1  {  // private instance variable  private double[] rainfall;  // constructor with no parameters  public RainfallForWeek1()  {  // create array for 7 rainfall readings and  // initialize instance variable rainfall  rainfall = new double[7];  }  // set rainfall reading for one day  public void SetRainfallForDay(int day, double reading)  {  rainfall[day] = reading; // day used as index  }  // get rainfall reading for one day  public double GetRainfallForDay(int day)  {  return rainfall[day]; // day used as index  }  } |

|  |
| --- |
| using System;  public class RainfallForWeekTest1  {  public static void Main()  {  RainfallForWeek1 record = new RainfallForWeek1();  // set rainfall readings for 7 days  // days of week represented as 0 to 6  record.SetRainfallForDay(0, 5.2);  record.SetRainfallForDay(1, 0.0);  record.SetRainfallForDay(2, 8.0);  record.SetRainfallForDay(3, 2.7);  record.SetRainfallForDay(4, 0.0);  record.SetRainfallForDay(5, 1.9);  record.SetRainfallForDay(6, 0.0);  // get rainfall readings for the 7 days  // days of week represented as 0 to 6  Console.WriteLine("Day\tRainfall");  Console.WriteLine("0\t{0:F1}", record.GetRainfallForDay(0));  Console.WriteLine("1\t{0:F1}", record.GetRainfallForDay(1));  Console.WriteLine("2\t{0:F1}", record.GetRainfallForDay(2));  Console.WriteLine("3\t{0:F1}", record.GetRainfallForDay(3));  Console.WriteLine("4\t{0:F1}", record.GetRainfallForDay(4));  Console.WriteLine("5\t{0:F1}", record.GetRainfallForDay(5));  Console.WriteLine("6\t{0:F1}", record.GetRainfallForDay(6));  }  } |

4. The following program contains a RainfallForWeek2 class and a RainfallForWeekTest2 class. The RainfallForWeek2 class has the following:

* An instance variable which is an array to store rainfall readings for 7 days of a week.
* A static variable representing a class constant for the number of days in a week
* A constructor with an array parameter representing the rainfall readings for a week.
* An instance method with one parameter representing a day. The method that returns the rainfall reading from the array for that day.

Examine and run the program.

|  |
| --- |
| public class RainfallForWeek2  {  // private instance variable  private double[] rainfall;  // public static variable - class constant  // represents constant for number of days per week  public static int DAYS\_PER\_WEEK = 7;  // constructor with an array parameter for 7 rainfall readings  public RainfallForWeek2(double[] readings)  {  // initialise rainfall to array parameter  rainfall = readings;  }  // get rainfall reading for one day  public double GetRainfallForDay(int day)  {  return rainfall[day]; // day used for index  }  } |

|  |
| --- |
| using System;  public class RainfallForWeekTest2  {  public static void Main()  {  double[] readings = { 5.2, 0.0, 8.0, 2.7, 0.0, 1.9, 0.0 };  RainfallForWeek2 record = new RainfallForWeek2(readings);  Console.WriteLine("Day\tRainfall");  // days of week represented as 0 to 6  for (int day = 0; day < RainfallForWeek2.DAYS\_PER\_WEEK; day++)  {  Console.WriteLine("{0}\t{1:F1}", day, record.GetRainfallForDay(day));  }  }  } |

**Part B**

1. Create a class named Student. The data items are the name and 4 test scores. Use an array to store the test scores (*array of* *double*).

* Provide a constructor that has two parameters: a string to represent the name and an array of double to represent the test scores.
* Provide an instance method named ComputeAverage() that finds and returns the average of the 4 test scores (as a double value).

public class Student

{

private string name;

public string Name

{

get { return name; }

}

private double[] scores;

public Student(string theName, double[] theScores)

{

theName = name;

scores = theScores;

}

public double ComputeAverage()

{

double total = 0.0;

foreach (double score in scores)

total += score;

return (total / 4);

}

Test the Student class using the following test class:

|  |
| --- |
| using System;  public class StudentTest1  {  public static void Main()  {  double[] scores = { 90.0, 80.0, 80.0, 90.0 };  Student student;  student = new Student("Alice Smith", scores);  Console.WriteLine("Average for {0} is {1:F1}", student.Name,  student.ComputeAverage());  }  } |

1. Modify the Student class in Question 1 to include a second constructor that has only 1 parameter which represents the name of the student.

The constructor initializes the name and creates an array that will be able to store 4 test scores.

Include an instance method named SetATestScore() which has 2 parameters. The first parameter represents the test number (with values ranging from 0 to 3). The second parameter represents the test score (with values ranging from 0 to 100).

Test your modified Student class using the following test class:

|  |
| --- |
| using System;  public class StudentTest2  {  public static void Main()  {  Student student = new Student("Joe");  student.SetATestScore(0, 100.0);  student.SetATestScore(1, 90.0);  student.SetATestScore(2, 90.0);  student.SetATestScore(3, 100.0);  Console.WriteLine("Average for {0} is {1:F1}", student.Name,  student.ComputeAverage());  }  } |

public class Student

{

private string name;

public string Name

{

get { return name; }

}

private double[] scores;

public Student(double[] theScores)

{

scores = theScores;

}

public Student(string tName)

{

name = tName;

scores = new double[4];

}

public void SetATestScore(int number, double score)

{

scores[number] = score;

}

public double ComputeAverage()

{

double total = 0.0;

foreach (double score in scores)

total += score;

return (total / 4);

}

1. Modify the Student class in Question 2 to add the following instance methods:

* FindLowest() – to find and return the lowest test score for a student
* FindHighest() – to find and return the highest test score for a student

Test your modified Student class using the following Main() method:

public double FindLowest()

{

return scores.Min();

}

public double FindHighest()

{

return scores.Max();

}

|  |
| --- |
| using System;  public class StudentTest3  {  public static void Main()  {  Student student = new Student("Joe Binder");  student.SetATestScore(0, 100.0);  student.SetATestScore(1, 90.0);  student.SetATestScore(2, 90.0);  student.SetATestScore(3, 100.0);  Console.WriteLine("Lowest test score for {0} is {1}", student.Name,  student.FindLowest());  Console.WriteLine("Highest test score for {0} is {1}", student.Name,  student.FindHighest());  }  } |

1. Modify the class in Student Question 3 to add an instance method GetATestScore() that has 1 parameter representing the test number (with values ranging from 0 to 3). The method returns the test score for that test number.

Test your modified Student class using the following Main() method:

public double GetATestScore(int test)

{

return scores[test];

}

|  |
| --- |
| using System;  public class StudentTest3  {  public static void Main()  {  Student student = new Student("Joe Binder");  student.SetATestScore(0, 100.0);  student.SetATestScore(1, 90.0);  student.SetATestScore(2, 90.0);  student.SetATestScore(3, 100.0);  for (int testNumber = 0; testNumber < 4; testNumber++)  Console.WriteLine("Score for test {0} is {1:F1}",  testNumber, student.GetATestScore(testNumber));  }  } |

1. Complete the 2 static methods given below. The first static method inputs student details, then creates and returns a Student object. The second static method receives a Student object as a parameter and displays the student details.

|  |
| --- |
| using System;  public class StudentApp  {  public static void Main()  {  Student student = StudentApp.GetStudentData();  StudentApp.DisplayStudentData(student);  }  public static Student GetStudentData()  {  // ask user to input name & 4 test scores  // create object for student by calling second constructor  // use object created, call/invoke the SetaScore method  // return the object  }  public static void DisplayStudentData(Student aStudent)  {  // complete this method  }  } |

public class Student

{

private string name;

public string Name

{

get { return name; }

}

private double[] scores;

public Student(string tName)

{

name = tName;

scores = new double[4];

}

public Student(double[] theScores)

{

scores = theScores;

}

public double GetATestScore(int test)

{

return scores[test];

}

public void SetATestScore(int number, double score)

{

scores[number] = score;

}

public double ComputeAverage()

{

double total = 0.0;

foreach (double score in scores)

total += score;

return (total / 4);

}

public class StudentApp

{

public static void Main()

{

//student invokes get student data

Student student = StudentApp.GetStudentData();

//display get student data

StudentApp.DisplayStudentData(student);

}

public static Student GetStudentData()

{

Console.WriteLine("Input name:");

string iname = Console.ReadLine();

Student data = new Student(iname);

string imark;

for(int i =0; i< data.scores.Length;i++)

{

Console.WriteLine("Test Mark 1-4:");

imark = Console.ReadLine();

data.scores[i] = Convert.ToDouble(imark);

data.SetATestScore(i, data.scores[i]);

}

return data;

}

public static void DisplayStudentData(Student aStudent)

{

// complete this method

Console.WriteLine("Details of Student:");

Console.WriteLine("{0} {1}",aStudent.name,aStudent.scores);

}

}

}

<http://www.softwareandfinance.com/CSharp/Student_Grading.html>

1. Complete the Main() method given below to create 3 Student objects that represents the following 3 students, stores them in array students, and then displays the names and average test score for each student.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *No.* | *Name* | *Test Score 1* | *Test Score 2* | *Test Score 3* | *Test Score 4* |
| 1 | Alice | 100 | 90 | 80 | 100 |
| 2 | Bobby | 60 | 70 | 80 | 90 |
| 3 | Charlene | 70 | 60 | 80 | 90 |

|  |
| --- |
| using System;  public class StudentApp  {  public static void Main()  {  Student [] students = new Student[3];  // complete this method  students[0] = new Student("Alice");  students[1] = new Student("Bobby");  students[2] = new Student("Charlene");  }  } |