**Object-Oriented Application Development**

**Practical 5**

**Part A**

1. Examine and run the following program which creates an array of objects.

|  |
| --- |
| public class Car  {  private string regNumber;  public string RegNumber  {  get { return regNumber; }  }  public Car(string number)  {  regNumber = number;  }  } |

|  |
| --- |
| using System;  public class CarArray  {  public static void Main()  {  // create an array that can store 3 Car  // objects  Car [] cars = new Car[3];  // create a Car object  Car c1 = new Car("ABC1234");  // store it into the array  cars[0] = c1;  // create 2 more Car objects and store them  // into the array  cars[1] = new Car("ABC5678");  cars[2] = new Car("ABC9999");  // Display car details - version 1  for (int i = 0; i < cars.Length; i++)  {  Car car = cars[i];  Console.WriteLine(car.RegNumber);  }  // Display car details – version 2  for (int i = 0; i < cars.Length; i++)  {  Console.WriteLine(cars[i].RegNumber);  }  // Display car details - version 3  foreach (Car aCar in cars)  {  Console.WriteLine(aCar.RegNumber);  }  }  } |

1. Examine and run the following program which passes objects between methods.

|  |
| --- |
| public class Car  {  private string regNumber;  public string RegNumber  {  get { return regNumber; }  }  public Car(string number)  {  regNumber = number;  }  } |

|  |
| --- |
| using System;  public class OneCarApp  {  public static void Main()  {  // method returns Car object    Car car = CreateCar();  // pass Car object to method  DisplayCarInfo(car);  }  public static Car CreateCar()  {  Console.Write("Enter reg. number: ");  string number = Console.ReadLine();    Car car = new Car(number);  return car;  }  public static void DisplayCarInfo(Car car)  {  Console.WriteLine(car.RegNumber);  }  } |

1. Examine and run the following program which creates an array of objects and passes objects between methods.

|  |
| --- |
| public class Car  {  private string regNumber;  public string RegNumber  {  get { return regNumber; }  }  public Car(string number)  {  regNumber = number;  }  } |

|  |
| --- |
| using System;  public class ManyCarsApp  {  public static void Main()  {  // create array to store maximum of  // 10 Car objects  Car[] carArray = new Car[10];  // keep track of number of Car objects  // created  int count = 0;  string answer;  do  {  Car car = CreateCar();  carArray[count] = car;  count++;  Console.Write("Anymore cars? ");  answer = Console.ReadLine();  } while (answer == "yes");  for (int i = 0; i < count; i++)  {  DisplayCarInfo(carArray[i]);  }  }  public static Car CreateCar()  {  Console.Write("Enter reg. number: ");  string number = Console.ReadLine();    Car car = new Car(number);  return car;  }  public static void DisplayCarInfo(Car car)  {  Console.WriteLine(car.RegNumber);  }  } |

1. Examine and run the following program which contains relationship between classes.

|  |
| --- |
| public class Country  {  private string name;  public string Name  {  get { return name; }  }  private string continent;  public string Continent  {  get { return continent; }  }    public Country(string aName, string aContinent)  {  name = aName;  continent = aContinent;  }  } |

|  |
| --- |
| public class Tourist  {  private string name;  public string Name  {  get { return name; }  }  private Country country; //‘has-a’ relationship  public Country Country  {  get { return country; }  }    public Tourist(string aName, Country aCountry)  {  name = aName;  country = aCountry;  }  } |

|  |
| --- |
| using System;  public class TouristTest  {  public static void Main(string[] args)  {  Country japan = new Country("Japan", "Asia");  Tourist kenjo = new Tourist("Kenjo", japan);  Console.WriteLine("{0} is from {1}", kenjo.Name, kenjo.Country.Name);  }  } |

**Part B**

1. Consider the Country and Tour classes given below.

|  |
| --- |
| public class Country  {  private string name;  public string Name  {  get { return name; }  }  private string continent;  public string Continent  {  get { return continent; }  }  public Country(string aName, string aContinent)  {  name = aName;  continent = aContinent;  }  } |

|  |
| --- |
| public class Tour  {  private string code;  public string Code  {  get { return code; }  }  private Country destination;  public Country Destination  {  get { return destination; }  }  private int numberOfDays;  public int NumberOfDays  {  get { return numberOfDays; }  }  public Tour(string aCode, Country aDestination, int days)  {  code = aCode;  destination = aDestination;  numberOfDays = days;  }  } |

1. Draw a class diagram to represent the classes and their relationship. Include details as given in the code.
2. Write the code for a class that contains a Main() method which does the following:

* Create an array that can be used to store 4 Country objects.
* Create 4 Country objects representing the following countries and store them into the array:

|  |  |
| --- | --- |
| *Country* | *Continent* |
| Korea | Asia |
| United Kingdom | Europe |
| Canada | North America |
| South Africa | Africa |

* Display the details of all the Country objects in the array.

public class Country

{

private string name;

private string continent;

public string cname

{

get { return name; }

}

public string Continent

{

get { return continent; }

}

public Country(string name, string continent)

{

this.name = name;

this.continent = continent;

}

public class Array

{

public static void Main()

{

Country[] country = new Country[4];

Country korea = new Country("Korea","Asia");

Country uk = new Country("United Kingdom","Europe");

Country canada = new Country("Canada","North America");

Country africa = new Country("South Africa","Africa");

country[0] = korea;

country[1] = uk;

country[2] = canada;

country[3] = africa;

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

{

Country count = country[i];

Console.WriteLine(country[i].cname + country[i].Continent);

}

}

}

}

1. Write the code for a class that contains a Main() method which does the following:

* Create Country objects and Tour objects representing the following tours and stores them into an array.

|  |  |  |
| --- | --- | --- |
| *Tour Code* | *Country* | *Number of Days* |
| K1 | Korea | 5 |
| K2 | Korea | 7 |
| U1 | United Kingdom | 10 |
| U2 | United Kingdom | 12 |

* Display the details of code, destination country name, and number of days of all the tours.
* public class Country
* {
* private string name;
* public string Name
* {
* get { return name; }
* }
* private string continent;
* public string Continent
* {
* get { return continent; }
* }
* public Country(string aName, string aContinent)
* {
* name = aName;
* continent = aContinent;
* }
* }
* public class Tour
* {
* private string code;
* public string Code
* {
* get { return code; }
* }
* private string destination;
* public string Destination
* {
* get { return destination; }
* }
* private int numberOfDays;
* public int NumberOfDays
* {
* get { return numberOfDays; }
* }
* public Tour(string aCode, string aDestination, int days)
* {
* code = aCode;
* destination = aDestination;
* numberOfDays = days;
* }
* }
* public class Array
* {
* public static void Main()
* {
* Tour[] travel = new Tour[4];
* Tour k1 = new Tour("K1", "Korea", 5);
* Tour k2 = new Tour("K2", "Korea", 7);
* Tour u1 = new Tour("U1", "United Kingdom", 10);
* Tour u2 = new Tour("U2", "United Kingdom", 12);
* travel[0] = k1;
* travel[1] = k2;
* travel[2] = u1;
* travel[3] = u2;
* for (int i = 0; i < travel.Length; i++)
* {
* Tour t = travel[i];
* Console.WriteLine("Tour code is" + travel[i].Code);
* Console.WriteLine("Destination is" + travel[i].Destination);
* Console.WriteLine("Number of Days is" + travel[i].NumberOfDays);
* }
* }

1. Consider the Passenger and Flight classes given below.

|  |
| --- |
| public class Passenger  {  private string name;  public string Name  {  get { return name; }  }  private string passportNumber;  public string PassportNumber  {  get { return passportNumber; }  }  public Passenger(string aName, string aPassportNumber)  {  name = aName;  passportNumber = aPassportNumber;  }  } |

|  |
| --- |
| public class Flight  {  private string code;  public string Code  {  get { return code; }  }  private string departureAirport;  public string DepartureAirport  {  get { return departureAirport; }  }  private string arrivalAirport;  public string ArrivalAirport  {  get { return arrivalAirport; }  }  private int numberOfPassengers;  public int NumberOfPassengers  {  get { return numberOfPassengers; }  }  private Passenger[] passengerList;  public Passenger[] PassengerList  {  get { return passengerList; }  }  private static readonly int MAX\_NUMBER\_OF\_PASSENGERS = 300;  public Flight(string aCode, string departure, string arrival)  {  code = aCode;  departureAirport = departure;  arrivalAirport = arrival;  numberOfPassengers = 0;  passengerList = new Passenger[MAX\_NUMBER\_OF\_PASSENGERS];  }  public void RecordAPassenger(Passenger passenger)  {  // to be completed (refer to part b)  if (numberOfPassengers < MAX\_NUMBER\_OF\_PASSENGERS];  {  passengerList [numberOfPassengers] = passenger;  numberOfPassengers++;  }  }  } |

1. Draw a class diagram to represent the classes and their relationship. Include details as given in the code.
2. Complete the code for the RecordAPassenger() method in the Flight class which has a Passenger object parameter. The method stores the Passenger object into the array.

3. Consider the following classes:

|  |
| --- |
| public class Participant  {  private string name;  public string Name  {  get { return name; }  set { name = value; }  }    private string icNumber;  public string IcNumber  {  get { return icNumber; }  set { icNumber = value; }  }  public Participant(string aName, string aIcNumber)  {  Name = aName;  IcNumber = aIcNumber;  }  } |

|  |
| --- |
| public class Seminar  {  private string title;  public string Title  {  get { return title; }  set { title = value; }  }    private string code;  public string Code  {  get { return code; }  set { code = value; }  }  private Participant[] participantList;  public Participant[] ParticipantList  {  get { return participantList; }  }  private int numberOfParticipants;  public int NumberOfParticipants  {  get { return numberOfParticipants; }  }    private static readonly int MAX\_NUMBER\_OF\_PARTICIPANTS = 30;    public Seminar(string aCode, string aTitle)  {  Code = aCode;  Title = aTitle;  numberOfParticipants = 0;  participantList = new Participant[MAX\_NUMBER\_OF\_PARTICIPANTS];  }  public void Register(Participant participant)  {  if (numberOfParticipants < MAX\_NUMBER\_OF\_PARTICIPANTS)  {  participantList[numberOfParticipants] = participant;  numberOfParticipants++;  }  }  } |

1. Write a class with a Main() method that inputs data for a seminar and creates a Seminar object.

public class SeminarTest

{

public static void Main()

{

String s = Console.ReadLine();

String a = Console.ReadLine();

Seminar seminar = new Seminar(s, a);

Console.WriteLine("Seminar i: " + s + a);

}

}

1. Extend the Main() method in (a) to handle registration of participants for the seminar. The program inputs data for each participant. After all the participants are registered, the program displays details of all registered participants.

string p1 = Console.ReadLine();

string ic1 = Console.ReadLine();

Participant p = new Participant(p1, ic1);

Console.WriteLine(" Participant are " + p1 + ic1);

1. Consider the following Controller class that contains an array that can store Seminar objects.

|  |
| --- |
| public class Controller  {  private Seminar[] seminarList;  public Seminar[] SeminarList  {  get { return seminarList; }  }    private int numberOfSeminars;  public int NumberOfSeminars  {  get { return numberOfSeminars; }  }  private static int MAX\_NUMBER\_OF\_SEMINARS = 10;  public Controller()  {  seminarList = new Seminar[MAX\_NUMBER\_OF\_SEMINARS];  numberOfSeminars = 0;  }    public void AddNewSeminar(Seminar seminar)  {  seminarList[NumberOfSeminars] = seminar;  numberOfSeminars++;  }    } |

Write a class with a Main() method that uses a Controller object to record details of seminars and their participants. After all the seminar and participant details are entered and recorded, the program displays the details of each seminar and the participants registered for the seminar.