|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scheme of Valuation/Answer Key**  (Scheme of evaluation (marks in brackets) and answers of problems/key) | | | | | |
| **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**  SIXTH SEMESTER B.TECH DEGREE EXAMINATION, JUNE 2019 | | | | | |
| **Course Code: EC312** | | | | | |
| **Course Name:Object Oriented Programming** | | | | | |
| Max. Marks: 100 | | |  | Duration: 3 Hours | |
| **PART A** | | | | | |
|  |  | ***Answer any two full questions, each carries 15 marks*** | | | Marks |
| 1 | a) | **Features (any 4)-4 marks**   * Inheritance. * Polymorphism. * Data Hiding. * Encapsulation. * Overloading. * Reusability.   **Advantages-3 marks**  Modular structure  Reuse  Easier to maintain  Faster development  Better productivity | | | (7) |
|  | b) | **Problem statement-3 marks**  In C++ multilevel inheritance, a class has more than one parent class. For example, if we take animals as a base class then mammals are the derived class which has features of animals and then humans are the also derived class that is derived from sub-class mammals which **inherit** all the features of mammals  **Program implementation-5 marks-(Out of which 2 marks may be given for syntax)** | | | (8) |
| 2 | a) | class room  {  intwidth,length;  void setvalue(int w, l)------------------------ Error-( int w, int l)  {  width=w; length=l}  };  void main()  {  room classroom;  classroom.setvalue(12,13);-----------------Error- setvalue is not in public visibility mode  ….  }  Identification of errors-2 marks each  Correction-1.5 marks each | | | (7) |
|  | b) | Define object -3 marks , example -1 mark  When class is defined, only the specification for the object is defined; no memory or storage is allocated. To use the data and access functions defined in the class, you need to create objects.   1. class Student { 2. public: 3. int id;//data member (also instance variable) 4. string name;//data member(also instance variable) 5. };   Student s1; //creating an object of Student | | | (4) |
|  | c) | Need of abstract class-2 marks, example-2 marks  The purpose of an abstract class (often referred to as an ABC) is to provide an appropriate base class from which other classes can inherit. Abstract classes cannot be used to instantiate objects and serves only as an interface. Attempting to instantiate an object of an abstract class causes a compilation error.  class Base  {     int x;  public:      virtual void fun() = 0;      intgetX() { return x; }  }; | | | (4) |
| 3 | a) | **Use of constructors -1**  It is used to initialize the various data elements of different objects with different values when they are created. A constructor is a special type of member function that initialises an object automatically when it is created.  **use of destructors-1**  Destructors are usually used to deallocate memory and do other cleanup for a class object and its class members when the object is destroyed.  **Example including 3 types of constructors-5**  #include <iostream>  using namespace std;    class Point {  private:      int x, y;    public:      // Parameterized Constructor      Point(int x1, int y1)      {          x = x1;          y = y1;      }      Point() //default constructor      {          x = 0;          y = 0;      }      Point()      {          x = 1;          y = 1;      }        intgetX()      { return x; }      intgetY()      { return y;  }  };    int main()  {      // Constructor called      Point p1(10, 15); // Access values assigned by constructor      cout<< "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();        return 0;  } | | | (7) |
|  | b) | 1. Program without friend function -4 marks-(Out of which 2 marks can be given for correct syntax) 2. Program with friend function-4 marks-(Out of which 2 marks can be given for correct syntax) | | | (8) |
| **PART B** | | | | | |
| ***Answer any two full questions, each carries 15 marks*** | | | | | |
| 4 | a) | **Run time polymorphism-3 marks**  This type of polymorphism is achieved by Function Overriding. Function overriding on the other hand occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.  **Compile time polymorphism-4 marks for 2 methods(function overloading, operator overloading)**  This type of polymorphism is achieved by function overloading or operator overloading.  **Function Overloading**: When there are multiple functions with same name but different parameters then these functions are said to be **overloaded**. Functions can be overloaded by **change in number of arguments** or/and **change in type of arguments.**  Operator overloading: C++ also provides option to overload operators. For example, we can make the operator (‘+’) for string class to concatenate two strings. We know that this is the addition operator whose task is to add two operands. So a single operator ‘+’ when placed between integer operands, adds them and when placed between string operands, concatenates them. | | | (7) |
|  | b) | Explain different forms of inheritance in Java program-4 marks   * Single Inheritance. * Multiple Inheritance (Through Interface) * Multilevel Inheritance. * Hierarchical Inheritance. * Hybrid Inheritance (Through Interface)   Write a program to illustrate single inheritance in Java.-4 marks(Out of which 2 marks can be given for correct syntax)  public class Shape  {  int length;  int breadth;  }  public class Rectangle extends Shape  {  int area;  public void calcualteArea()  {  area = length\*breadth;  }  public static void main(String args[])  {  Rectangle r = new Rectangle();  //Assigning values to Shape class attributes  r.length = 10;  r.breadth = 20;  //Calculate the area  r.calcualteArea();  System.out.println("The Area of rectangle of length \""  +r.length+"\" and breadth \""+r.breadth+"\" is \""+r.area+"\"");  }  } | | | (8) |
| 5 | a) | When do we make a virtual function pure? – 3 marks  A pure virtual function or pure virtual method is a virtual function that is required to be implemented by a derived class if the derived class is not abstract.Pure virtual methods typically have a declaration (signature) and no definition (implementation).  Example program-4 marks | | | (7) |
|  | b) | Program -8marks (Out of which 3 marks can be given for correct syntax) | | | (8) |
| 6 | a) | How can you create pointers to objects in C++ -3 marks  Accessing members of a class-4 marks  A pointer to a C++ class is done exactly the same way as a pointer to a structure and to access members of a pointer to a class you use the member access operator -> operator, just as you do with pointers to structures. | | | (7) |
|  | b) | Explanation of Layers of interaction of a Java program -4 marks  It is the organization of the project structure into four main categories: **presentation, application, domain, and infrastructure**. Each of the layers contains objects related to the particular concern it represents.  Explanation how Java achieve architecture neutrality-4 marks  Java is architecture neutral only because the JVM abstracts away the specifics of the particular machine where java code runs on.Java has different implementations of the JVM (called JRE) for different OS.So Java is architecture neutral but JVM is ultimately architecture dependent | | | (8) |
| **PART C** | | | | | |
| ***Answer any two full questions, each carries 20 marks*** | | | | | |
| 7 | a) | Figure of architecture-5 marks  Explanation -5 marks | | | (10) |
|  | b) | Explanation of Broadcast receivers -3 marks  Explanation of Content providers- 3 marks  Implemented in android program-4 marks | | | (10) |
| 8 | a) | Explain any five features of Android OS-2 marks each | | | (10) |
|  | b) | What is an activity in Android-4 marks  activity life cycle diagram-3 marks  Explanation-3 marks | | | (10) |
| 9 | a) | Steps – 12 marks | | | (12) |
|  | b) | File where permission, intent filters, receivers etc are set – 3 marks each  Description related to these points - 5 marks | | | (8) |
| \*\*\*\* | | | | | |