**CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Engineering/Computer Science & Engineering/ Information Technology

**Subject Name: Object Oriented Programming with C++**

**Semester: II**

**Subject Code: CE144**

**Academic year: 2020-21**

|  |  |
| --- | --- |
| **36.** | **What is the output of the following code:**   1. **Pointer to Objects**   **PROGRAM CODE :**  #include <iostream>  using namespace std;  class product  {  int code;  float price;  public:  void getdata(int a, float b)  {  code = a;  price = b;  }  void show()  {  cout << "Code: " << code << endl;  cout << "Price: " << price << endl;  }  };  int main()  {  product \*p = new product;  product \*d = p;  int x, i;  float y;  cout << "Input code and price for product: ";  cin >> x >> y;  p->getdata(x, y);  d->show();  }  **OUTPUT:**     1. **this pointer**   **PROGRAM CODE :**  #include <iostream>  using namespace std;  class student  {  int roll\_no;  float age;  public:  student(int r, float a)  {  roll\_no = r;  age = a;  }  student &greater(student &x)  {  if (x.age >= age)  return x;  else  return \*this;  }  void display()  {  cout << "Roll No " << roll\_no << endl;  cout << "Age " << age << endl;  }  };  int main()  {  student s1(23, 18), s2(30, 20), s3(45, 16);  student s = s1.greater(s3);  cout << "Elder Person is :" << endl;  s.display();  }  **OUTPUT:**     1. **Pointers to Derived Objects**   **PROGRAM CODE :**  #include <iostream>  using namespace std;  class BC  {  public:  int b;  void show()  {  cout << "b = " << b << endl;  }  };  class DC : public BC  {  public:  int d;  void show()  {  cout << "b = " << b << endl;  cout << "d = " << d << endl;  }  };  int main()  {  BC \*bptr;  BC base;  bptr = &base;  bptr->b = 100;  cout << "bptr poins to base objects" << endl;  bptr->show();  DC derived;  bptr = &derived;  bptr->b = 200;  /\*bptr->b = 300;\*/ // wont work  cout << "bptr now points to derived object" << endl;  bptr->show();  DC \*dptr;  dptr = &derived;  dptr->d = 300;  cout << "Dptr is derived type pointer" << endl;  dptr->show();  return 0;  }  **OUTPUT:**    **CONCLUSION:** In this practical we learn about pointer to object, this pointer and pointer to derived objects. |