**Jaypee University of Engineering and Technology**

**B. Tech. (CSE) - II Semester**

**Object Oriented Programming (18B11CI211)**

**Tutorial – 5(Constructors & Destructors)**

|  |  |
| --- | --- |
|  | Q1. Create two classes the first holds customer data- specifically, a customer number and zip code. The second, a class for cities, holds the city name, state, and zip code. Additionally, each class contains a constructor that takes arguments to set the field values. Create a friend function that displays a customer number and the customer’s city, state, and zip code. Write a brief main() function to test the classes and friend function.  Q2. Predict the output of following programs:  b.#include<iostream>  using namespace std;  class Point {      Point() { cout << "Constructor called"; }  };  int main(){     Point t1;     return 0; }  a.#include <iostream>  using namespace std;    int i;   class A {  public:      ~A()      {         i=10;     }  };  int foo() {      i=3;      A ob;      return i;  }  int main() {      cout << foo() << endl;      return 0;  } |
|  |  |
|  | |  |  | | --- | --- | |  |  | |
| |  |  | | --- | --- | | d.#include <iostream>  using namespace std;  class A  {      int id;      static int count;  public:      A() {          count++;          id = count;          cout << "constructor for id " << id << endl;      }      ~A() {          cout << "destructor for id " << id << endl;      }  };  int A::count = 0;  int main() {      A a[3];      return 0;  } |  |  |  |  | | --- | --- | | 1. e. #include <iostream> 2. using namespace std; 3. class Box 4. { double width; 5. public: 6. friend void printWidth( Box box ); 7. void setWidth( double wid ); 8. }; 9. void Box::setWidth( double wid )   { width = wid; } | 1. void printWidth( Box box ) 2. { 3. box.width = box.width \* 2;   cout << "Width of box : " << box.width; }   1. int main( ) 2. { Box box; 3. box.setWidth(10.0); 4. printWidth( box );   return 0;  } | | |  |  | | --- | --- | |  | c.#include<iostream>  using namespace std;  class X {  public:      int x; };  int main(){      X a = {10};      X b = a;      cout << a.x << " " << b.x;      return 0;} |  |  |  | | --- | --- | | e. #include <iostream>  using namespace std;  class A{  private:  A(){  cout << "constructor of A\n";  }  friend class B;  };  class B{  public:  B(){  A a1;  cout << "constructor of B\n";  }  };    int main(){  B b1;  return 0;  } |  | |
| Q.3 | Like constructors, can there be more than one destructors in a class? **(A)** Yes **(B)** No |
| Q.4. | Can destructors be private in C++? **(A)** Yes **(B)** No |
| Q.5 | What is the use of this pointer?  **(A)** When local variable’s name is same as member’s name, we can access member using this pointer. **(B)** To return reference to the calling object **(C)** Can be used for chained function calls on an object **(D)** All of the above |
|  |
|  |  |
| Q.6. | Which rule will not affect the friend function?  (A) private and protected members of a class cannot be accessed from outside (B) private and protected member can be accessed anywhere (C) both a & b (D) None of the mentioned |
|  |  |

|  |
| --- |
|  |