PRACTICAL – 6

PROGRAM -1

AIM- POLYGON PROGRAM USING ABSTRACTION

CODE-

|  |
| --- |
| from abc import ABC,abstractmethod  # This program will print the number of sides of different polygons.  print("HARSH D")  class Polygon(ABC):    """Represents a polygon."""    @abstractmethod    def Side(self):      """Prints the number of sides of the polygon."""      pass  class Triangle(Polygon):    """Represents a triangle."""    def Side(self):      """Prints that the polygon has 3 sides."""      print("I have 3 sides")  class Pentagon(Polygon):    """Represents a pentagon."""    def Side(self):      """Prints that the polygon has 5 sides."""      print("I have 5 sides")  class Hexagon(Polygon):    """Represents a hexagon."""    def Side(self):      """Prints that the polygon has 6 sides."""      print("I have 6 sides")  class Square(Polygon):    """Represents a square."""    def Side(self):      """Prints that the polygon has 4 sides."""      print("I have 4 sides")  object=Triangle()  object.Side()  object=Pentagon()  object.Side()  object=Hexagon()  object.Side()  object=Square()  object.Side() |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -2

AIM- SERVICE PROVIDER BASED ON BILLING METHOD BASED ON ABSTRACTION

CODE:-

|  |
| --- |
| from abc import ABC, abstractmethod  class ServiceProvider(ABC):      def \_\_init\_\_(self, name, per\_minute\_charge):          self.name = name          self.per\_minute\_charge = per\_minute\_charge      @abstractmethod      def calculate\_bill(self, minutes):          pass      def get\_lowest\_bill(self, minutes):          providers = [self, Jio("Jio", 0.75), Airtel("Airtel", 1.00), Vodafone("Vodafone", 0.90)]          lowest\_bill = float('inf')          lowest\_provider = None          for provider in providers:              bill = provider.calculate\_bill(minutes)              if bill < lowest\_bill:                  lowest\_bill = bill                  lowest\_provider = provider.name          return lowest\_provider, lowest\_bill  class Jio(ServiceProvider):      def calculate\_bill(self, minutes):          return minutes \* self.per\_minute\_charge  class Airtel(ServiceProvider):      def calculate\_bill(self, minutes):          return minutes \* self.per\_minute\_charge  class Vodafone(ServiceProvider):      def calculate\_bill(self, minutes):          return minutes \* self.per\_minute\_charge  def main():      jio\_provider = Jio("Jio", 0.75)      airtel\_provider = Airtel("Airtel", 1.00)      vodafone\_provider = Vodafone("Vodafone", 0.90)      minutes = int(input("Enter the number of minutes used: "))      lowest\_provider, lowest\_bill = jio\_provider.get\_lowest\_bill(minutes)      print(f"\nBill details for {jio\_provider.name}: ${jio\_provider.calculate\_bill(minutes):.2f}")      print(f"Bill details for {airtel\_provider.name}: ${airtel\_provider.calculate\_bill(minutes):.2f}")      print(f"Bill details for {vodafone\_provider.name}: ${vodafone\_provider.calculate\_bill(minutes):.2f}")      print(f"Lowest bill provider: {lowest\_provider}")      print(f"Lowest bill amount: ${lowest\_bill:.2f}")  if \_\_name\_\_ == "\_\_main\_\_":      main() |

OUTPUT:-

|  |
| --- |
|  |