```php
<?php
/**
* This class provides functions to deal with the database.
*/

class Database
{
    // private variables
    private $host; # hostname
    private $db; # name of database

    private $user; # name of user
    private $pwd; # password of user. Can be set via XAMMP

    private $con; # Connection
    private $port = 3306; # default port for MySQL (XAMPP control panel)

    /**
    * Initialize with parameters to connect to the database
    */
    function __construct($host, $db, $user, $pwd, $port = 3306, $autoconnect = true)
    {
        $this->host = $host;
        $this->db = $db;
        $this->user = $user;
        $this->pwd = $pwd;
        $this->port = $port;
        if($autoconnect)
        {
            $this->open();
        }
    }

    /**
    * Connect to db
    */
    function open()
    {
        $this->con = new mysqli($this->host, $this->db, $this->user, $this->pwd,
        $this->port);
    }
    /**
    * Insert statement
    */
    function insert()
    {
        if ( empty( $table ) || empty( $data ) )
        {
            return false;
        }
        // Connect to the database
        $db = $this->open();

        // Cast $data and $format to arrays
        $data = (array) $data;
        $format = (array) $format;

        // Build format string
        $format = implode('', $format);
        $format = str_replace('%', '', $format);

        list( $fields, $placeholders, $values ) = $this->prep_query($data);

        // Prepend $format onto $values
        array_unshift($values, $format);

        // Prepary our query for binding
        $stmt = $db->prepare("INSERT INTO {$table} ({$fields}) VALUES
        ({$placeholders})");

        // Dynamically bind values
        call_user_func_array( array( $stmt, 'bind_param'),
```

```
            $this->ref_values($values));
71
72          // Execute the query
73          $stmt->execute();
74
75          // Check for successful insertion
76          if ( $stmt->affected_rows ) {
77              return true;
78          }
79
80          return false;
81      }
82
83      /**
84      * Execute your query
85      */
86      function query( $query )
87      {
88          return $this->con->query( $query ); # &query is the sql statement
89      }
90
91      /**
92       * Close connection
93      */
94      function close()
95      {
96          $this->con->close();
97      }
98
99      private function prep_query($data, $type='insert')
100     {
101         // Instantiate $fields and $placeholders for looping
102         $fields = '';
103         $placeholders = '';
104         $values = array();
105
106         // Loop through $data and build $fields, $placeholders, and $values
107         foreach ( $data as $field => $value )
108         {
109             $fields .= "{$field},";
110             $values[] = $value;
111
112             if ( $type == 'update')
113             {
114                 $placeholders .= $field . '=?,';
115             }
116             else
117             {
118                 $placeholders .= '?,';
119             }
120         }
121         // Normalize $fields and $placeholders for inserting
122         $fields = substr($fields, 0, -1);
123         $placeholders = substr($placeholders, 0, -1);
124
125         return array( $fields, $placeholders, $values );
126     }
127 }
128 ?>
```