What is Descriptive Programming?   
Descriptive Programming is a method of performing operation on the object which is not there in Object Repository. We can also use programmatic descriptions to perform the same operation on several objects with certain identical properties, or to perform an operation on an object whose properties match a description that you determine dynamically during the run session

When to use Descriptive Programming?   
Here there are some situations   
1.      If the application  is having Dynamic Objects  
        OR:- Difficult to handle Dynamic Objects using Object Repository  
2.      When we have more objects to perform operations  
        OR:- The performance will decrease if object repository is having huge number of objects.  
3.      If the application  is having objects that are adding in the Run Time  
        OR:- We can’t add objects to Object Repository in run time.  
4.      If we need to start Automation before Build Release  
        OR:- There is no application to create Object Repository.  
5.      If Application is having similar type of objects or similar name objects  
        OR:- Object Repository will create multiple objects with same description unnecessarily.  
6.      Big Team Size   
        OR:- Shared Object Repository is not changeable by multiple persons at a time.  
        Maintenance becomes harder if all the team members have created their own object repositories.

Q. What are test objects? Test objects are objects created and maintained by QTP in its object repository. Each test object has a name and a set of properties to uniquely identify the actual object in the application.   
  
Q. how does QTP identifies the objects during the script run? During script run QuickTest searches for a run-time object that has the similar description of thetest object in the object repository.   
  
Q. How can i check the properties of an object in an application without using checkpoints? For Example how will you check whether a button is enabled? Answer: GetROProperty method is used to retrieve the properties of the object in an application.It is very useful method and can be used to retrieve almost any of the property that can be seenwhile spying the object with object spy. For Example‗To get whether button is enabled or not.

**2) What are the types object Repositories in QTP.**

QTP Supports 2 types of Object Repository

1) **Shared**Object Repository (also called **Global**)

2)**Per-Action**Object Repository, (also called **Local**)

Per-Action Object Repository is used by default. The extension for Per-Action repository is **".mtr"**.

Shared Object Repository is preferable while dealing with dynamic objects which are called in multiple tests. The extension is **".tsr"**

**3) Can we call QTP test from another test using scripting. Suppose there are 4 tests and I want to call these tests in a main script. Is this possible in QTP?**

**Yes**.  You can call 4 or even more scripts in your tests.For this, first you will need to make the Actions in the corresponding scripts **re-usable**.Then from the destination script you can make calls to these re-usable actions.

**4) What is action split and the purpose of using this in QTP?**

Action split is to divide an existing action into two parts.The purpose is to divide actions based on their functionality to **improve code re-use**.

**5) How will you handle Java tree in QTP ?**

Foremost you will select **Java Add - In** and launch QTP. Next step record operations on the Java Tree. If you face an issue while recording, you can select Tools > Object Identification > Java, tree object and make changes in mandatory and assistive properties to enable identification.

**Tip**: You can base you answer on similar lines for any other object of any environment. For example : If the question is how will check SAP checkbox , You say , first I will select SAP Add in ... and so on.

**6) Explain how QTP identifies object ?**

QTP identifies any GUI Object based on its corresponding **properties**.  While recording, QTP will identify and store peculiar properties (as defined in the Object Identification settings) in the object repository of the GUI object . At run-time, QTP will compare the stored property values with the on-screen properties, to uniquely identify the GUI object.

Learn more about [Object Identification](http://www.guru99.com/quick-test-professional-qtp-tutorial-6.html)

**7) How many types of recording modes in QTP? Which will be used when ?**

QTP supports 3 types of recording modes

1**. Normal**mode also called Contextual

2. **Low-level**recording mode

3.**Analog**mode

**Normal Mode**: It is the default recording mode and takes full advantage of QTP's Test Object Model. **It recognizes objects regardless of their position on -screen**. This is the preferred mode of recoding and is used for most of the automation activities.

**Low-level recording mode**: This mode records the **exact x,y co-ordinates** of your mouse operations. It is helpful in testing hashmaps. It is useful for recording objects not identified by normal mode of QTP.

**Analog mode**:This mode records **exact mouse and keyboard "movements"** you perform in relation to the screen / application window. This mode is useful for the operation such as drawing a picture, recording signature., drag and drop operations.

Learn more about [Recording Modes in QTP](http://www.guru99.com/quick-test-professional-qtp-tutorial-26.html)

**8) How will you  call from one action to another action ?**

We can call an action in 2 ways

**1) Call to copy of Action**. - In this ,the Action Object Repository , Script and Datable will be **copied**to the destination Test Script.  
**2)** **Call to Existing Action**. - In this,  Object Repository , Script and Datable  will NOT be copied but a **call (reference)** would be made to the Action in the source script.

**9) What are Virtual Objects?**

Your application may **contain objects that behave like standard objects but are not recognized by QTP**. You can **define these objects as virtual objects and map them to standard classes**, such as a button or a check box. QTP emulates the user's action on the virtual object during the run session. In the test results, the virtual object is displayed as though it is a standard class object.

For example, suppose you want to record a test on a Web page containing a bitmap that the user clicks. The bitmap contains several different hyperlink areas, and each area opens a different destination page. When you record a test, the Web site matches the coordinates of the click on the bitmap and opens the destination page.

To enable QTP to click at the required coordinates during a run session, you can define a virtual object for an area of the bitmap, which includes those coordinates, and map it to the button class. When you run a test, QTP clicks the bitmap in the area defined as a virtual object so that the Web site opens the correct destination page.

**10) How to perform Cross platform testing and Cross browser testing using QTP? Can u explain giving some example?**

You will need to create **separate Actions which take care of different OS and Browsers**

**Cross Platform Testing:**  
  
Using the Built in Environment Variable you can dig up the OS information.

Eg. Platform = Environment("OS"). Then based on the Platform you need to call the actions which you recorded on that particular platform.  
  
**Cross Browser Testing:**  
  
Using this code  Eg. Browser("Core Values").GetROProperty("version") you can extract the Browser and its correspondin version. Ex: Internet Explorer 6 or Netscape 5. Based on this value you call the actions which are relevant to that browser.

**11) What is logical name of the object?**

Logical name is a **name  given by  QTP** while creating an object in the repository to uniquely identify it from other objects in the application. This name would be used by the QTP to map the object name in script with its corresponding description in the object repository. **Ex: Browser("Browser").Page("Guru99") Here Guru99 is the logical name of the object.**

**12) What is descriptive programming?**

Typically ,an object and its properties must be recorded in the Object Repository to enable QTP to perform action s on it.

**Using descriptive programming , you do not store the object and its property values in the Object repository** but mention the *property value pair*directly in the script.

The idea behind descriptive programming is not bypass the object repository but help recogonize dynamic objects.

Learn more about [Descriptive Programming](http://www.guru99.com/quick-test-professional-qtp-tutorial-32.html)

**13)What are the properties you would use for identifying a browser & page when using descriptive programming ?**

You can use the **name**property

ex: Browser("name:="xxx"").page("name:="xxxx"").....

OR   
  
We can also use the property **"micClass"**.   
ex: Browser("micClass:=browser").page("micClass:=page")....

**14)Can we record an application running on a remote machine using QTP ?**

**Yes**.you can record remote application provided you are accessing application through the local browser not via remoter like citrix.

If you are still unable to record it is advisable install QTP and application, on the same machine

**15) Explain the keyword CreateObject with an example.**

Creates and returns a reference to an Automation object

**SYNTAX:** CreateObject(servername.typename [, location])

Arguments   
servername: **Required**. The name of the application providing the object.   
typename :  **Required**. The type or class of the object to create.   
location :  **Optional**. The name of the network server where the object is to be created.

**Example** : Set IE = CreateObject("InternetExplorer.Application")

**16) Can you switch between Per-Action and Shared Object Repository ? If yes how ?**

**Yes** .We can switch. Go to Test--->Settings--->Resources. Here  you have an option to choose repositories.

**17) What is Object Spy ? How to Use it ?**

Object Spy helps in **determining**the **run & test time object properties & methods**of the application under test.

You can access object spy directly from the toolbar or from the Object Repository Dialog Box.

It is very useful during Descriptive Programming

Learn more about [Object Spy](http://www.guru99.com/quick-test-professional-qtp-tutorial-31.html)

**18) When ordinal identifiers alone can make an object unique then why they are not given top priority? Why it is first mandatory and next assistive. Why we cannot go for ordinal identifiers directly?**

Consider the following -

a) If two **objects**are **overlapped**on each other than **location**based object recognition will **fail**.

b) If only **index**based recognition is used your script will work but script **execution time will increase**.

Hence mandatory and assistive properties are used.

**19) What is the file extension of the code file in QTP?**

Code file extension is *script****.mts***

**20) Explain in brief about the QTP Automation Object Model.**

QTP Automation Object model **deals with** **Automation of QTP itself**. Almost all configuration and functionality provided by QTP is represented by QTP's Automation Object Model . Almost all dialog boxes in QTP have a corresponding automation object which can set or retrieved using the corresponding properties or methods in the Automation Object Model.QTP Automation Objects can be used along with standard VB programming elements like iterative loops or conditional statements to help you design a script of choice.

**21) What is the use of Text output value in QTP?**

Text Output values enable you to **capture text**appearing on the application under test during run-time.

If parameterized, text output values will capture values appearing in each iteration which would be stored in the run-time data table for further analysis.

**22) What is Step Generator?**

Step Generator enables use to **Add Test Steps** in your script. Using step generator you can add steps to your script without actually recording it.

**23) How to make QTP understand the difference amongst the  same type of objects .Suppose there are 5 check boxes in a page and I have to choose the 2nd one, how to do that through script?**

You can use **ordinal identifiers lik**e **index**along with a little **descriptive programming** for object recognition.

Watch a [video](http://www.guru99.com/quick-test-professional-qtp-tutorial-33.html) of this example.

**24) What is Test Fusion Report ?.**

Test Fusion Report , displays all aspects of a test run and is organized in a **Tree format**.

It gives details of each step executed for all iterations.

It also gives Run-time data table, Screen shots and movie of the test run if opted.

**25) How can you handle exceptions in QTP?**

In QTP Exceptional handling is done by using

**a. Recovery Scenarios.  
b. Using “On Error” statement**

In Recovery scenario you have to define.  
1. Triggered Events.   
2. Recovery steps.  
3. Post Recovery Test-Run.

At Script Level you can use the On Error Resume Next and On Error Go to 0 statement.

1. **What are the Features & Benefits of Quick Test Pro (QTP 8.0)?** - Operates stand-alone, or integrated into Mercury Business Process Testing and Mercury Quality Center. Introduces next-generation zero-configuration Keyword Driven testing technology in Quick Test Professional 8.0 allowing for fast test creation, easier maintenance, and more powerful data-driving capability. Identifies objects with Unique Smart Object Recognition, even if they change from build to build, enabling reliable unattended script execution. Collapses test documentation and test creation to a single step with Auto-documentation technology. Enables thorough validation of applications through a full complement of checkpoints.
2. **How to handle the exceptions using recovery scenario manager in QTP?** - There are 4 trigger events during which a recovery scenario should be activated. A pop up window appears in an opened application during the test run: A property of an object changes its state or value, A step in the test does not run successfully, An open application fails during the test run, These triggers are considered as exceptions.You can instruct QTP to recover unexpected events or errors that occurred in your testing environment during test run. Recovery scenario manager provides a wizard that guides you through the defining recovery scenario. Recovery scenario has three steps: 1. Triggered Events 2. Recovery steps 3. Post Recovery Test-Run
3. **What is the use of Text output value in QTP?** - Output values enable to view the values that the application talks during run time. When parameterized, the values change for each iteration. Thus by creating output values, we can capture the values that the application takes for each run and output them to the data table.
4. **How to use the Object spy in QTP 8.0 version?** - There are two ways to Spy the objects in QTP: 1) Thru file toolbar, In the File Toolbar click on the last toolbar button (an icon showing a person with hat). 2) True Object repository Dialog, In Object repository dialog click on the button object spy. In the Object spy Dialog click on the button showing hand symbol. The pointer now changes in to a hand symbol and we have to point out the object to spy the state of the object if at all the object is not visible. or window is minimized then, hold the Ctrl button and activate the required window to and release the Ctrl button.
5. **How Does Run time data (Parameterization) is handled in QTP?** - You can then enter test data into the Data Table, an integrated spreadsheet with the full functionality of Excel, to manipulate data sets and create multiple test iterations, without programming, to expand test case coverage. Data can be typed in or imported from databases, spreadsheets, or text files.
6. **What is keyword view and Expert view in QTP?** - Quick TestÃ¢â‚¬â„¢s Keyword Driven approach, test automation experts have full access to the underlying test and object properties, via an integrated scripting and debugging environment that is round-trip synchronized with the Keyword View. Advanced testers can view and edit their tests in the Expert View, which reveals the underlying industry-standard VBScript that Quick Test Professional automatically generates. Any changes made in the Expert View are automatically synchronized with the Keyword View.
7. **Explain about the Test Fusion Report of QTP?** - Once a tester has run a test, a Test Fusion report displays all aspects of the test run: a high-level results overview, an expandable Tree View of the test specifying exactly where application failures occurred, the test data used, application screen shots for every step that highlight any discrepancies, and detailed explanations of each checkpoint pass and failure. By combining Test Fusion reports with Quick Test Professional, you can share reports across an entire QA and development team.
8. **Which environments does QTP support?** - Quick Test Professional supports functional testing of all enterprise environments, including Windows, Web,..NET, Java/J2EE, SAP, Siebel, Oracle, PeopleSoft, Visual Basic, ActiveX, mainframe terminal emulators, and Web services.
9. **What is QTP?** - Quick Test is a graphical interface record-playback automation tool. It is able to work with any web, java or windows client application. Quick Test enables you to test standard web objects and ActiveX controls. In addition to these environments, Quick Test Professional also enables you to test Java applets and applications and multimedia objects on Applications as well as standard Windows applications, Visual Basic 6 applications and.NET framework applications
10. **Explain QTP Testing process?** - Quick Test testing process consists of 6 main phases:
11. **Create your test plan** - Prior to automating there should be a detailed description of the test including the exact steps to follow, data to be input, and all items to be verified by the test. The verification information should include both data validations and existence or state verifications of objects in the application.
12. **Recording a session on your application** - As you navigate through your application, Quick Test graphically displays each step you perform in the form of a collapsible icon-based test tree. A step is any user action that causes or makes a change in your site, such as clicking a link or image, or entering data in a form.
13. **Enhancing your test** - Inserting checkpoints into your test lets you search for a specific value of a page, object or text string, which helps you identify whether or not your application is functioning correctly. NOTE: Checkpoints can be added to a test as you record it or after the fact via the Active Screen. It is much easier and faster to add the checkpoints during the recording process. Broadening the scope of your test by replacing fixed values with parameters lets you check how your application performs the same operations with multiple sets of data. Adding logic and conditional statements to your test enables you to add sophisticated checks to your test.
14. **Debugging your test** - If changes were made to the script, you need to debug it to check that it operates smoothly and without interruption.
15. **Running your test on a new version of your application** - You run a test to check the behavior of your application. While running, Quick Test connects to your application and performs each step in your test.
16. **Analyzing the test results** - You examine the test results to pinpoint defects in your application.
17. **Reporting defects** - As you encounter failures in the application when analyzing test results, you will create defect reports in Defect Reporting Tool.
18. **Explain the QTP Tool interface.** - It contains the following key elements: Title bar, displaying the name of the currently open test, Menu bar, displaying menus of Quick Test commands, File toolbar, containing buttons to assist you in managing tests, Test toolbar, containing buttons used while creating and maintaining tests, Debug toolbar, containing buttons used while debugging tests. Note: The Debug toolbar is not displayed when you open Quick Test for the first time. You can display the Debug toolbar by choosing View — Toolbars — Debug. Action toolbar, containing buttons and a list of actions, enabling you to view the details of an individual action or the entire test flow. Note: The Action toolbar is not displayed when you open Quick Test for the first time. You can display the Action toolbar by choosing View — Toolbars — Action. If you insert a reusable or external action in a test, the Action toolbar is displayed automatically. Test pane, containing two tabs to view your test-the Tree View and the Expert View ,Test Details pane, containing the Active Screen. Data Table, containing two tabs, Global and Action, to assist you in parameterizing your test. Debug Viewer pane, containing three tabs to assist you in debugging your test-Watch Expressions, Variables, and Command. (The Debug Viewer pane can be opened only when a test run pauses at a breakpoint.) Status bar, displaying the status of the test
19. **How does QTP recognize Objects in AUT?** - Quick Test stores the definitions for application objects in a file called the Object Repository. As you record your test, Quick Test will add an entry for each item you interact with. Each Object Repository entry will be identified by a logical name (determined automatically by Quick Test), and will contain a set of properties (type, name, etc) that uniquely identify each object. Each line in the Quick Test script will contain a reference to the object that you interacted with, a call to the appropriate method (set, click, check) and any parameters for that method (such as the value for a call to the set method). The references to objects in the script will all be identified by the logical name, rather than any physical, descriptive properties.
20. **What are the types of Object Repositories in QTP?** - Quick Test has two types of object repositories for storing object information: shared object repositories and action object repositories. You can choose which type of object repository you want to use as the default type for new tests, and you can change the default as necessary for each new test. The object repository per-action mode is the default setting. In this mode, Quick Test automatically creates an object repository file for each action in your test so that you can create and run tests without creating, choosing, or modifying object repository files. However, if you do modify values in an action object repository, your changes do not have any effect on other actions. Therefore, if the same test object exists in more than one action and you modify an objectÃ¢â‚¬â„¢s property values in one action, you may need to make the same change in every action (and any test) containing the object.
21. **Explain the check points in QTP?** - A checkpoint verifies that expected information is displayed in an Application while the test is running. You can add eight types of checkpoints to your test for standard web objects using QTP. A page checkpoint checks the characteristics of an Application. A text checkpoint checks that a text string is displayed in the appropriate place on an Application. An object checkpoint (Standard) checks the values of an object on an Application. An image checkpoint checks the values of an image on an Application. A table checkpoint checks information within a table on a Application. An Accessibilityy checkpoint checks the web page for Section 508 compliance. An XML checkpoint checks the contents of individual XML data files or XML documents that are part of your Web application. A database checkpoint checks the contents of databases accessed by your web site
22. **In how many ways we can add check points to an application using QTP?** - We can add checkpoints while recording the application or we can add after recording is completed using Active screen (Note : To perform the second one The Active screen must be enabled while recording).
23. **How does QTP identify objects in the application?** - QTP identifies the object in the application by Logical Name and Class.
24. **What is Parameterizing Tests?** - When you test your application, you may want to check how it performs the same operations with multiple sets of data. For example, suppose you want to check how your application responds to ten separate sets of data. You could record ten separate tests, each with its own set of data. Alternatively, you can create a parameterized test that runs ten times: each time the test runs, it uses a different set of data.
25. **What is test object model in QTP?** - The test object model is a large set of object types or classes that Quick Test uses to represent the objects in your application. Each test object class has a list of properties that can uniquely identify objects of that class and a set of relevant methods that Quick Test can record for it. A test object is an object that Quick Test creates in the test or component to represent the actual object in your application. Quick Test stores information about the object that will help it identify and check the object during the run session.
26. **What is Object Spy in QTP?** - Using the Object Spy, you can view the properties of any object in an open application. You use the Object Spy pointer to point to an object. The Object Spy displays the selected objectÃ¢â‚¬â„¢s hierarchy tree and its properties and values in the Properties tab of the Object Spy dialog box.
27. **What is the Diff between Image check-point and Bit map Check point?** - Image checkpoints enable you to check the properties of a Web image. You can check an area of a Web page or application as a bitmap. While creating a test or component, you specify the area you want to check by selecting an object. You can check an entire object or any area within an object. Quick Test captures the specified object as a bitmap, and inserts a checkpoint in the test or component. You can also choose to save only the selected area of the object with your test or component in order to save disk Space. For example, suppose you have a Web site that can display a map of a city the user specifies. The map has control keys for zooming. You can record the new map that is displayed after one click on the control key that zooms in the map. Using the bitmap checkpoint, you can check that the map zooms in correctly. You can create bitmap checkpoints for all supported testing environments (as long as the appropriate add-ins are loaded). Note: The results of bitmap checkpoints may be affected by factors such as operating system, screen resolution, and color settings.
28. **How many ways we can parameterize data in QTP?** –

There are four types of parameters: Test, action or component parameters enable you to use values passed from your test or component, or values from other actions in your test. Data Table parameters enable you to create a data-driven test (or action) that runs several times using the data you supply. In each repetition, or iteration, Quick Test uses a different value from the Data Table. Environment variable parameters enable you to use variable values from other sources during the run session. These may be values you supply, or values that Quick Test generates for you based on conditions and options you choose. Random number parameters enable you to insert random numbers as values in your test or component. For example, to check how your application handles small and large ticket orders, you can have Quick Test generate a random number and insert it in a number of tickets edit field.

1. **How do u do batch testing in WR & is it possible to do in QTP, if so explain?** - Batch Testing in WR is nothing but running the whole test set by selecting Run Test set from the Execution Grid. The same is possible with QTP also. If our test cases are automated then by selecting Run Test set all the test scripts can be executed. In this process the Scripts get executed one by one by keeping all the remaining scripts in Waiting mode.
2. **If I give some thousand tests to execute in 2 days what do u do?** - Adhoc testing is done. It Covers the least basic functionalities to verify that the system is working fine.
3. **What does it mean when a check point is in red color? what do u do?** - A red color indicates failure. Here we analyze the cause for failure whether it is a Script Issue or Environment Issue or a Application issue.
4. **What is Object Spy in QTP?** - Using the Object Spy, you can view the properties of any object in an open application. You use the Object Spy pointer to point to an object. The Object Spy displays the selected objectÃ¢â‚¬â„¢s hierarchy tree and its properties and values in the Properties tab of the Object Spy dialog box.
5. **What is the file extension of the code file & object repository file in QTP?** - Code file extension is.vbs and object repository is.tsr
6. **Explain the concept of object repository & how QTP recognizes objects?** - Object Repository: displays a tree of all objects in the current component or in the current action or entire test (depending on the object repository mode you selected). We can view or modify the test object description of any test object in the repository or to add new objects to the repository. Quicktest learns the default property values and determines in which test object class it fits. If it is not enough it adds assistive properties, one by one to the description until it has compiled the unique description. If no assistive properties are available, then it adds a special Ordinal identifier such as objects location on the page or in the source code.
7. **What are the properties you would use for identifying a browser & page when using descriptive programming?** - Name would be another property apart from title that we can use.
8. **Give me an example where you have used a COM interface in your QTP project?** - com interface appears in the scenario of front end and back end. for eg:if you r using oracle as back end and front end as VB or any language then for better compatibility we will go for an interface. of which COM will be one among those interfaces. Create object creates handle to the instance of the specified object so that we program can use the methods on the specified object. It is used for implementing Automation(as defined by Microsoft).
9. **Explain in brief about the QTP Automation Object Model.** - Essentially all configuration and run functionality provided via the Quick Test interface is in some way represented in the Quick Test automation object model via objects, methods, and properties. Although a one-on-one comparison cannot always be made, most dialog boxes in Quick Test have a corresponding automation object, most options in dialog boxes can be set and/or retrieved using the corresponding object property, and most menu commands and other operations have corresponding automation methods. You can use the objects, methods, and properties exposed by the Quick Test automation object model, along with standard programming elements such as loops and conditional statements to design your program.
10. **What is the default Add-ins in QTP?**
11. **Is QTP Supports UNI-CODE?**
12. **How you are developing the script? Using record and play back or manual?**
13. **What are Test Settings and Global Settings?**
14. **How many types of Objects are there in QTP?**
15. **\*What is test object?**
16. **\*What is Run Time Object?**
17. **\*Difference between Test Object and Run Time Object?**
18. **What is Action? How many types of Actions are there in QTP?**
19. **Difference between Copy to Action and Call to Action?**
20. **What is an Object Repository?**
21. **What are the types of Object Repositories? Which one is u using?**
22. **Have you faced any problems with Object Repository?**.
23. **Is it possible to merge two Object Repository files in QTP?**
24. **What is the difference between Per-Action and Shared?**Shared
25. **How QTP recognizes the Object?**
26. **If I change the object name in one action will it be updated in all the actions?Or not?**
27. **How to get the particular property value?**
28. **Is it possible to change the property value at runtime? How it is possible?**
29. **If I change the property value at runtime is it effect is Object Repository?**
30. **What is the parameterization? Give one example?**
31. **How many ways is to parameterize the value?**